EXHIBIT A

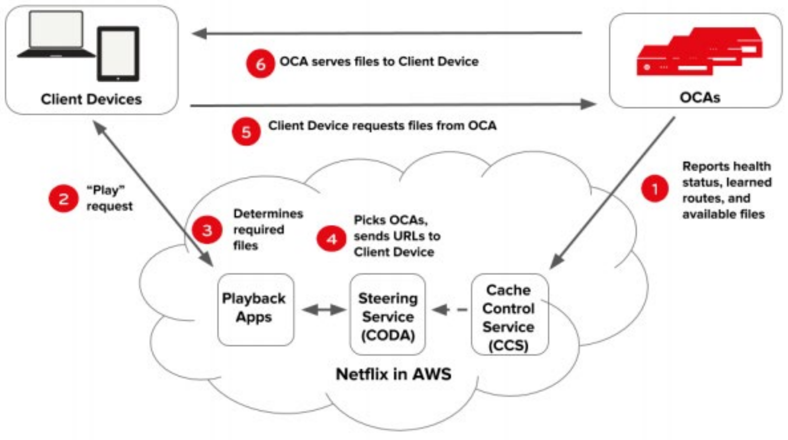
PLAINTIFF’S SUPPLEMENTAL INFRINGEMENT CONTENTIONS

June 13, 2023

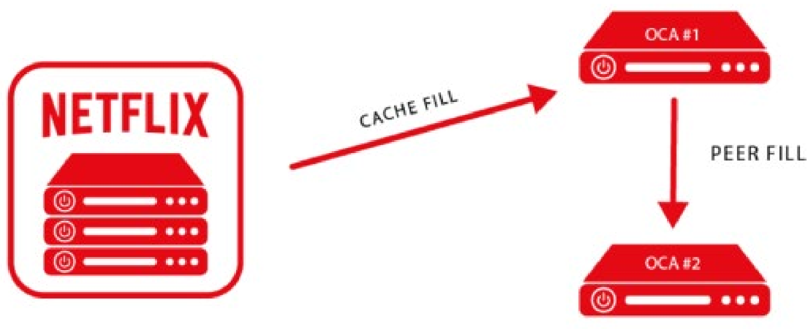
Claim Chart for U.S. Patent 8,495,167 - “Data communications networks, systems, methods and apparatus”

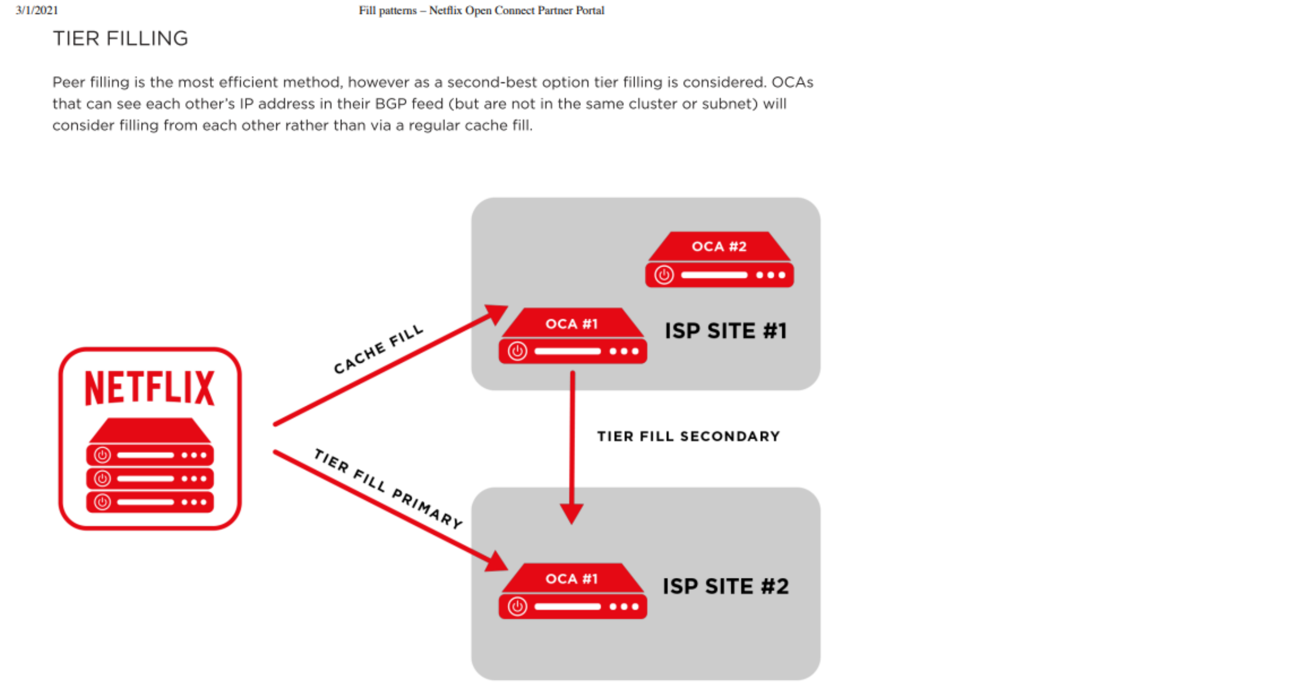
US Patent 8,495,167
Filing Date: Jul. 30, 2002
Priority Date: Jul 30, 2002

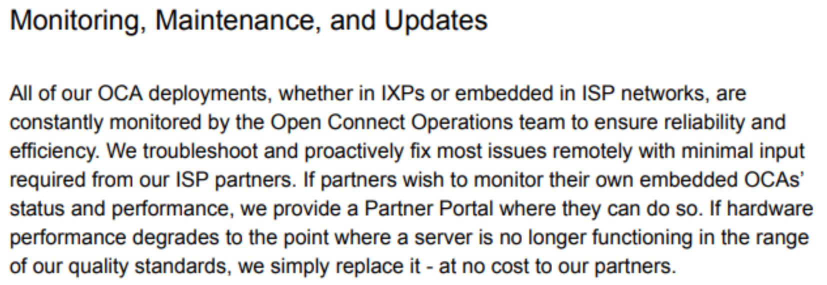
Claim Portion	‘167 Patent	Netflix
Claim 1		
[1a]	A data communication network comprising: a plurality of terminals; and	Netflix uses a system called **Open Connect** to deliver Netflix TV shows and movies to members world-wide. The building blocks of Open Connect are our suite of purpose-built **server appliances, called Open Connect Appliances (OCAs)**. *See* Open Connect Overview, p. 2. These are deployed directly inside ISP networks. Netflix provides the server hardware. The OCAs report to a Open Connect control plane to control fill behavior (adding new files to OCAs nightly) and to compute and/or store data. *See id.* p. 3-4. Accordingly, OCAs include both an input mechanism and display mechanism.

		 Client Devices OCAs 6 OCA serves files to Client Device 5 Client Device requests files from OCA 2 "Play" request 1 Reports health status, learned routes, and available files 3 Determines required files 4 Picks OCAs, sends URLs to Client Device Playback Apps Steering Service (CODA) Cache Control Service (CCS) Netflix in AWS Open Connect Appliances can be embedded in your ISP network. Embedded OCAs have the same capabilities as the OCAs that we use in our 60+ global data centers, and they are provided to qualifying ISP partners at no charge. Each embedded OCA deployment will offload a substantial amount of Netflix content traffic from peering or transport circuits. Multiple physical deployments can be distributed or clustered on a geographic or network basis to maximize local offload. Source: https://openconnect.netflix.com/en/#sample-architectures
[1b]	a main server adapted to manage selective retrieval of data from a first server by at least one target terminal selected from said plurality of terminals, said main server being	Netflix runs the operation of Open Connect from a Netflix application (CCS server) that is hosted in AWS. *See* Open Connect Overview, p. 4-5. In its global network, Netflix provides **data centers** such as an "S3" server **("first server")** housing **content ("data") on at lease one server**, and provides OCA users (such as ISP's) direct access to these data centers over the Internet that are housing the content. One or more of these **data centers** house a "**first server**" according to the claims. In deployment of Open Connect, Netflix provides Internet Service Providers with an OCA appliance direct "settlement-free interconnection (SFI)." The terminal OCA can "Connect via direct

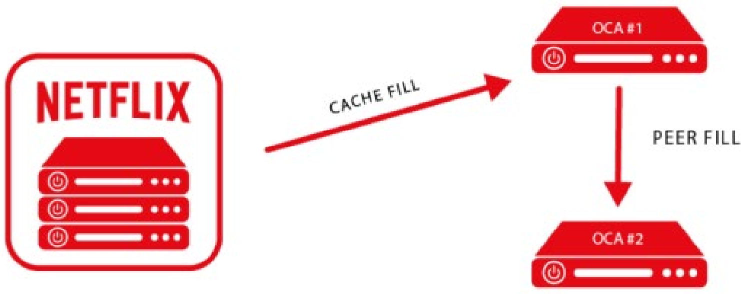
	distinct from said first server; and	Private Network Interconnect (PNI) or IXP-based SFI peering to Netflix Open Connect Appliances **in our data centers**." "Netflix has the ability to interconnect at a number of global data center facilities and public Internet Exchange fabrics as listed on our Peering Locations page. We openly peer with any network at IXP locations where we are mutually present and we consider private interconnection as appropriate." ISPs who do not currently participate in pubic peering might want to consider that a single IX port can support multiple peering sessions, providing **direct access to various content,** cloud, and network providers. Welcome to Open Connect, p. 3. Dkt 39 at p. 48. The following diagram also illustrates access from a target terminal (OCA #1, OCA #2) to a Netflix first server in "our data centers." OCAs in a cluster and on the same subnet can attempt peer filling from each other. There is also Tier filling where if in different ISP sites. deploymentguide.pdf (netflix.com)

		 https://openconnect.zendesk.com/hc/en-us/articles/360035618071 2/4

		 3/1/2021 Fill patterns – Netflix Open Connect Partner Portal TIER FILLING Peer filling is the most efficient method, however as a second-best option tier filling is considered. OCAs that can see each other's IP address in their BGP feed (but are not in the same cluster or subnet) will consider filling from each other rather than via a regular cache fill. NETFLIX CACHE FILL TIER FILL PRIMARY OCA #2 OCA #1 ISP SITE #1 TIER FILL SECONDARY OCA #1 ISP SITE #2 The "target terminal selected form said plurality of terminals" language of the claim is infringed by the health and performance monitoring and OCA target terminal selection process described below in Sec. [1e].
[1c]	a network information database containing terminal performance information, wherein	Netflix constantly measures and analyzes [OCA] performance and augment capacity as requirements evolve. *See* Open Connect Overview, p. 3. All OCA deployments are constantly monitored by the Open Connect Operations team to ensure reliability and efficiency. We troubleshoot and proactively fix most issues remotely with minimal input required from our ISP partners. *See* Open Connect Overview, p. 5. Additionally, OCAs periodically report health. *Id.* at 4.

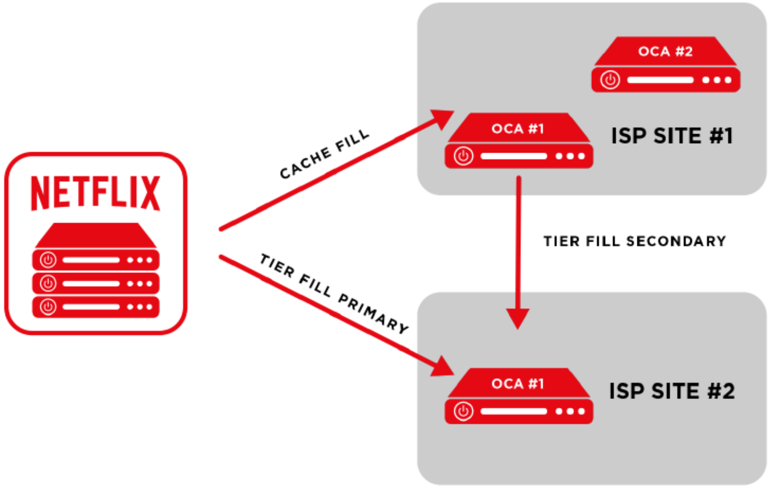
		 Monitoring, Maintenance, and Updates All of our OCA deployments, whether in IXPs or embedded in ISP networks, are constantly monitored by the Open Connect Operations team to ensure reliability and efficiency. We troubleshoot and proactively fix most issues remotely with minimal input required from our ISP partners. If partners wish to monitor their own embedded OCAs' status and performance, we provide a Partner Portal where they can do so. If hardware performance degrades to the point where a server is no longer functioning in the range of our quality standards, we simply replace it - at no cost to our partners. The main server on CCS is further adapted to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis of their relative response times. At https://netflixtechblog.com/netflix-and-fill-c43a32b490c0, Netflix wrote: The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title. The determination of the list takes into consideration several high-level factors: Title (content) availability — Does the fill source have the requested title stored? **Fill health — Can the fill source take on additional fill traffic?** A calculated route cost — Described in the next section. A fill escalation policy defines: 1. How many hops away an OCA can go to download content, and **how long it should wait before doing so** 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), **and how long it should wait before doing so** 3. Whether the OCA can go to S3, **and how long it should wait before doing so** (Emphasis added.)

		The CCS server monitors the “fill health” of each potential download locations, or fill sources that includes OCAs, and uses a fill escalation policy based upon response times to determine if the performance of the download location can take on additional traffic or not and how long the OCA should wait for a response before escalation to the next download locations. Saving this information to a database is common practice within network management.
[1d]	at least two of said terminals are adapted to act as relay servers for serving data retrieved from said first server to at least one target terminal; and wherein	A ”first server” or S3 is identified above in Sec. [1a] as a server within Netflix’ data centers, and at least two of the OCAs are adapted to act as relay servers for serving data retrieved from said first server to at least one OCA terminal. According to Netflix’ network architecture, OCAs in a cluster and on the same subnet can attempt peer filling content “cache fill” from the first server to OCA #1. OCA#1 then acts as a “relay server” to fill cached content to each other, from OCA#1 to OCA#2. OCA#2 becomes a “target terminal” in this example that is served data from the relay server (OCA#1) retrieved from the first server (data center). See https://openconnect.netflix.com/deploymentguide.pdf.

		 https://openconnect.zendesk.com/hc/en-us/articles/360035618071 2/4

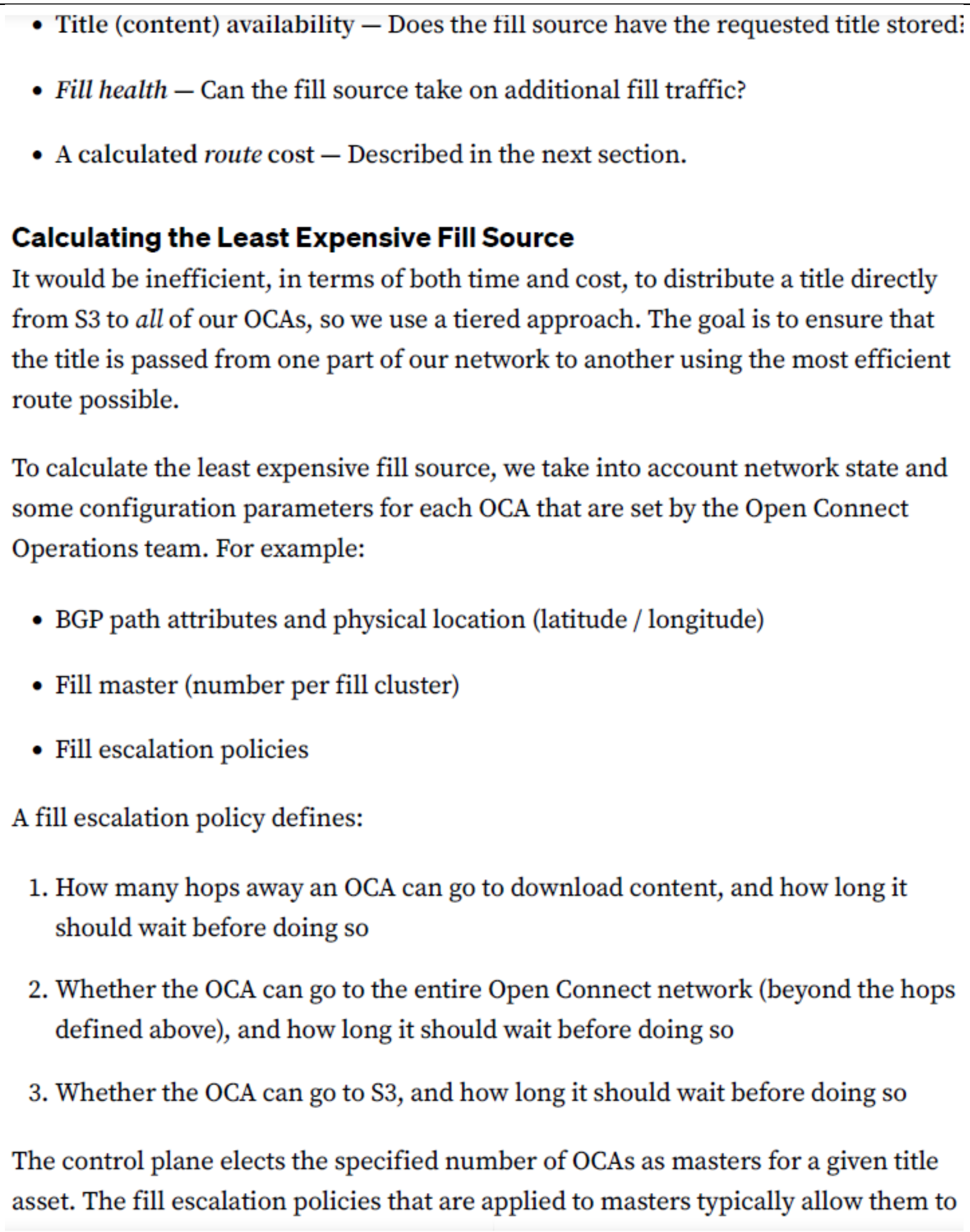
In another example, Netflix's first server (data center) can cache fill content to a terminal OCA#1, which then acts as a "relay server" to fill a target terminal (OCA#2) using "tier filling."

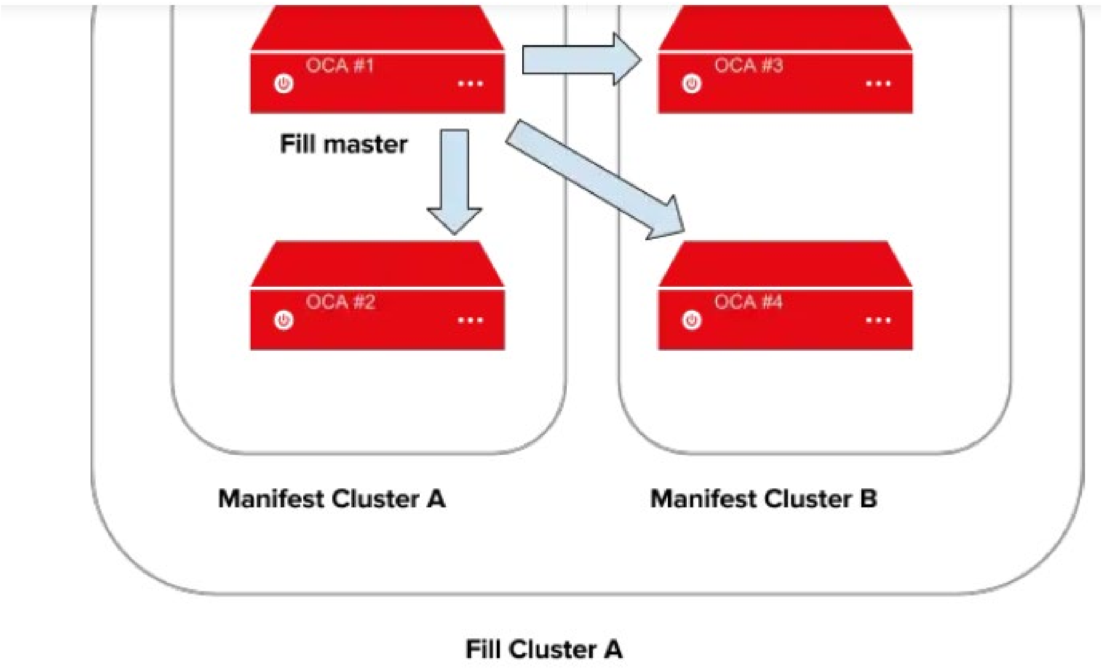
"Peer filling is the most efficient method, however as a second-best option tier filling is considered. OCAs that can see each other's IP address in their BGP feed (but are not in the same cluster or subnet) will consider filling from each other rather than via a regular cache fill."



See https://openconnect.netflix.com/deploymentguide.pdf.

In another example, https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states that there are "**The control plane elects the specified number of OCAs as masters. . . .**" OCAs can act as relay servers, or "masters" that target terminals can use to gain, or fill, content:

		 • Title (content) availability — Does the fill source have the requested title stored? • *Fill health* — Can the fill source take on additional fill traffic? • A calculated *route* cost — Described in the next section. **Calculating the Least Expensive Fill Source** It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to *all* of our OCAs, so we use a tiered approach. The goal is to ensure that the title is passed from one part of our network to another using the most efficient route possible. To calculate the least expensive fill source, we take into account network state and some configuration parameters for each OCA that are set by the Open Connect Operations team. For example: • BGP path attributes and physical location (latitude / longitude) • Fill master (number per fill cluster) • Fill escalation policies A fill escalation policy defines: 1. How many hops away an OCA can go to download content, and how long it should wait before doing so 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so 3. Whether the OCA can go to S3, and how long it should wait before doing so The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to

OCA #1
Fill master
OCA #2
OCA #3
OCA #4
Manifest Cluster A
Manifest Cluster B
Fill Cluster A

		When the second tier of OCAs complete their download, they report back their status, other OCAs can then fill from them, and so on. This process continues during the fill window. If there are titles being stored on an OCA that are no longer needed, they are put into a delete manifest and then deleted after a period of time that ensures we don't interrupt any live sessions. As the sun moves west and more members begin streaming, the fill window in this time zone ends, and the fill pattern continues as the fill window moves across other time zones — until enough of the OCAs in our global network that need to be able to serve this new title have it stored.

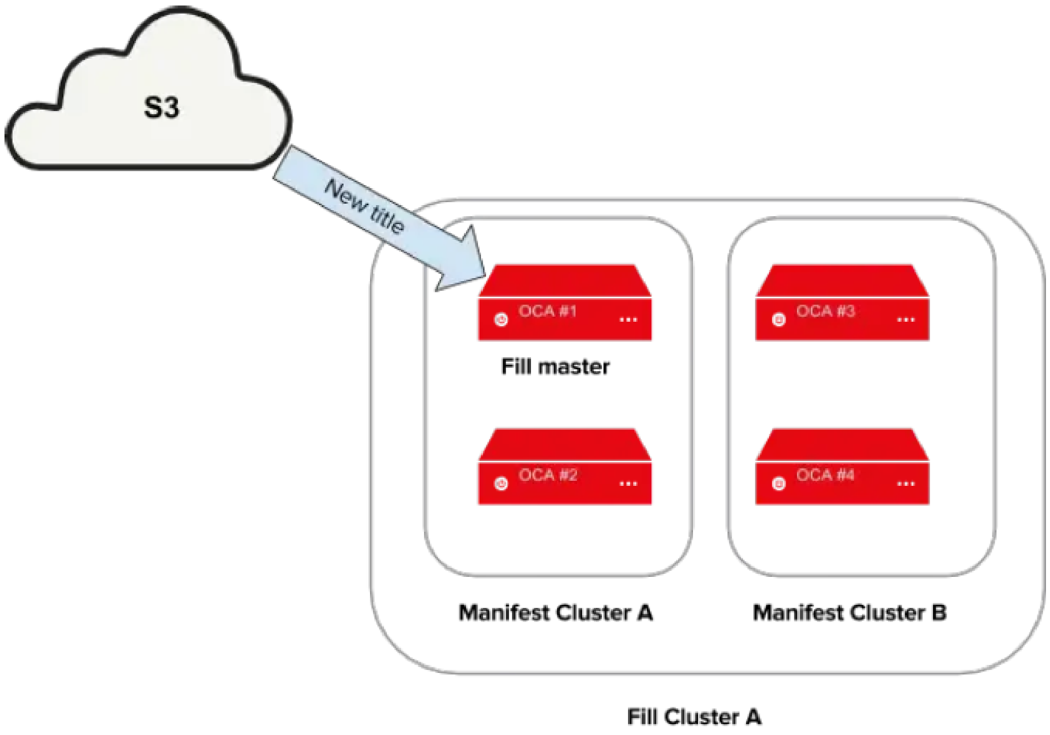
1. Peer fill: Available OCAs within the same manifest cluster or the same subnet

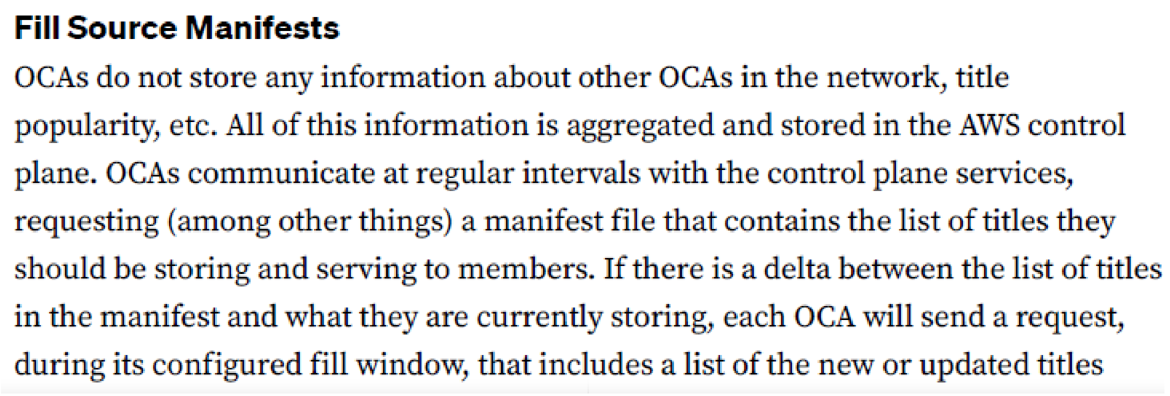
2. Tier fill: Available OCAs outside the manifest cluster configuration

3. Cache fill: Direct download from S3

Example Scenario

After the fill master OCA has completed its S3 download, it reports back to the control plane that it now has the title stored. The next time the other OCAs communicate with the control plane to request a fill source for this title, they are given the option to fill from the fill master.



[1e]	the main server is adapted to send transport requests direct to at least one first target terminal on the basis of said terminal performance information, and wherein the main server is further adapted to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis of their relative response times, and the first target terminal is adapted to act as relay server; and	All OCA deployments are constantly monitored to ensure reliability and efficiency. Netflix **makes use of non-peak bandwidth to download the vast majority of content updates to the OCAs in network during these configurable time windows. OCAs can also download updates from each other – minimizing significant usage of internet "backbone" capacity during the update cycle.** A "desired manifest" and when needed an "emergency manifest" are posted by Netflix to the CCS server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need.  **Fill Source Manifests** OCAs do not store any information about other OCAs in the network, title popularity, etc. All of this information is aggregated and stored in the AWS control plane. OCAs communicate at regular intervals with the control plane services, requesting (among other things) a manifest file that contains the list of titles they should be storing and serving to members. If there is a delta between the list of titles in the manifest and what they are currently storing, each OCA will send a request, during its configured fill window, that includes a list of the new or updated titles A "desired manifest" and when needed an "emergency manifest" are posted by Netflix to the CCS (Control Content Service) server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. Equivalently, these manifests are fetched by regularly from the CCS by each OCA terminal, which traverses the desired manifest or emergency manifest. The CCS server acts as an email inbox for manifests sent directly to the OCAs, where each OCA terminal is mandated to check regularly for manifests and download the manifests to the local OCA terminal hard drive.

		This desired manifest and emergency manifest and the download location data on the CCS are equivalent the "server is adapted to send transport requests direct to at least one first target terminal." The functions are substantially the same - to send a request from a main server (CCS) to a terminal (OCA) with instructions to download or "fill" content. The way this is accomplished is substantially the same. The claim states that the main server is adapted to send transport requests direct to at least one first target terminal over a network, and on the Open Connect Network, Netflix posts a desired manifest or emergency manifest to the CCS server that is intended for each OCA in the network. The results are substantially the same - to transmit commands from the main server/CCS server to a terminal/OCA to download content data in the manifest that needs to be retrieved. The control plane elects the specified number of OCAs as masters for a given title asset. https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 Thus, the CCS server selects OCA terminals as a download location for a given title asset. The main server on CCS is further adapted to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis of their relative response times. At https://netflixtechblog.com/netflix-and-fill-c43a32b490c0, Netflix wrote: The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title. The determination of the list takes into consideration several high-level factors: Title (content) availability — Does the fill source have the requested title stored? **Fill health — Can the fill source take on additional fill traffic?** A calculated route cost — Described in the next section. A fill escalation policy defines: 4. How many hops away an OCA can go to download content, and **how long it should wait before doing so**

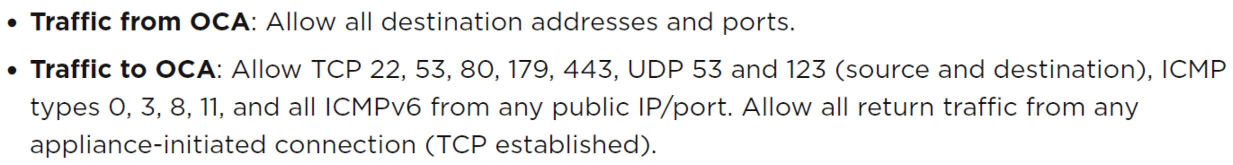
		5. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), **and how long it should wait before doing so** 6. Whether the OCA can go to S3, **and how long it should wait before doing so** (Emphasis added.) The CCS server monitors the "fill health" of each potential download locations, or fill sources that includes OCAs, and uses a fill escalation policy based upon response times to determine if the performance of the download location can take on additional traffic or not and how long the OCA should wait for a response before escalation to the next download locations. Thus, the manifest is assembled by the CCS server and download locations are assembled based on OCA performance information, and the manifest is sent to a given OCA. Therefore, the transport request is sent on a basis of said terminal performance information.
[1f]	wherein each such transport request includes details of data to be retrieved, the address of the first server from which the data is to be requested by the first target terminal, the addresses of at least one second target terminal to which the data from the first server to be relayed by the first target terminal and an indication of a	A "desired manifest" and when needed an "emergency manifest" are transport requests posted by Netflix to the CCS (Control Content Service) server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. these manifests are fetched by regularly from the CCS by each OCA terminal, which traverses the desired manifest or emergency manifest. OCAs then then query the CCS for location information files that list where each title on the desired manifest that is needed by an OCA can be downloaded. https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states a master OCA and at least one second OCA can be selected based on their relative performance: The CCS server information location list provides the address of a first server, called the "S3" server, for a download location and provides other download locations of a second and additional master OCAs: It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to all of our OCAs, so we use a tiered approach. . . . A fill escalation policy defines:

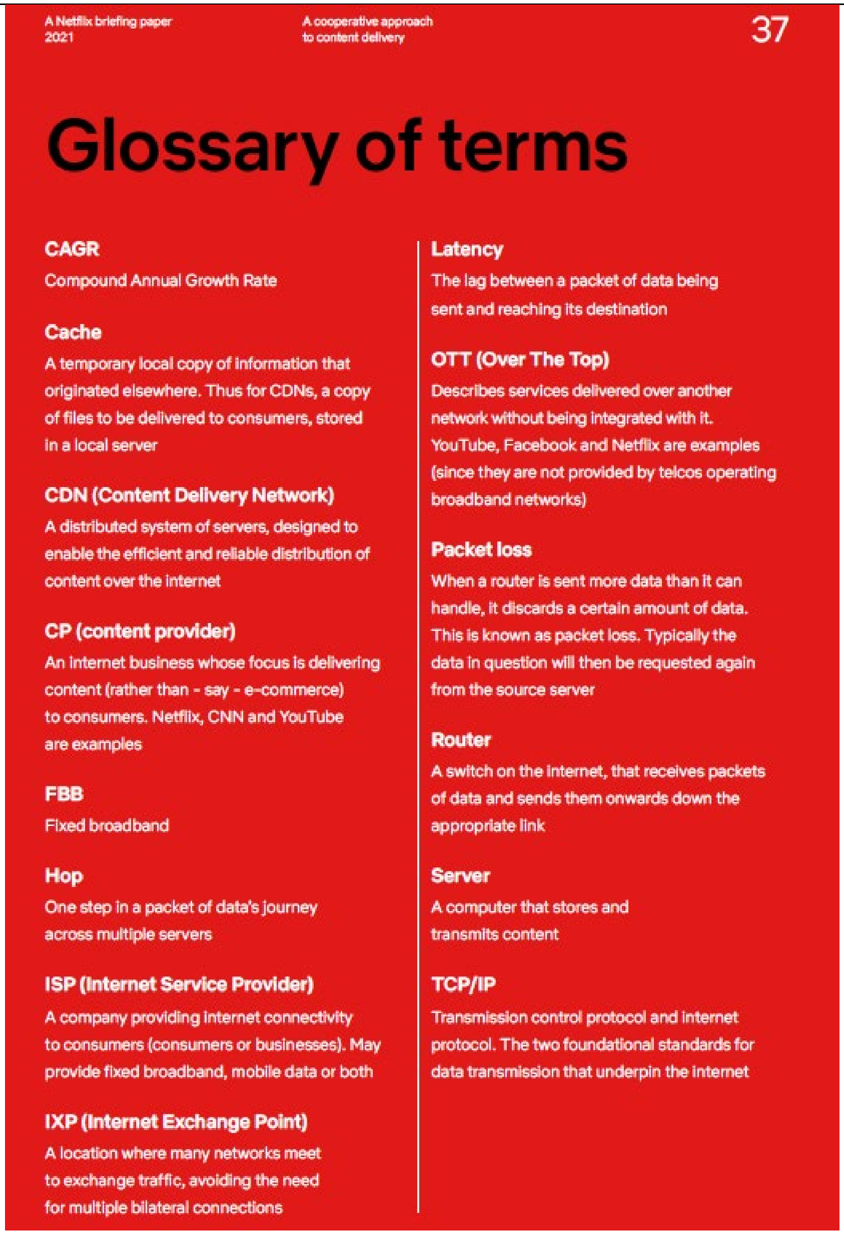
	relative performance of a further target terminal based on the terminal performance information stored in the network information database; and	1. How many hops away an OCA can go to download content, and how long it should wait before doing so 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so 3. Whether the OCA can go to S3, and how long it should wait before doing so The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to reach farther with less delay in order to grab that content and then share it locally with non-masters. As stated in [1e], the CCS monitors the “fill health” and performance of download locations, which is based on performance of the OCAs, to determine if that OCA will be selected as a download location or not. This desired manifest and emergency manifest along with the downloaded location information files and fill policy for master OCAs and a second (target terminal) OCA to fill from a (first terminal) OCA on the CCS are equivalent the “each such transport request includes details of data to be retrieved, the address of the first server from which the data is to be requested by the first target terminal, the addresses of at least one second target terminal to which the data from the first server to be relayed by the first target terminal.” The functions are substantially the same - to send a request from a main server (CCS) to a terminal (OCA) with instructions to download or “fill” content from specific master terminal/OCA addresses and includes an address of at least one second terminal/OCA. The way this is accomplished is substantially the same. The claim states that the main server is adapted to send transport requests that include download locations direct to at least one first target terminal over a network, and on the Open Connect Network, Netflix posts a desired manifest or emergency manifest to the CCS server along with download locations that are intended for each OCA to read on a regular basis. The results are substantially the same - to transmit commands from the main server/CCS server to a terminal/OCA to download content data in the manifest that needs to be retrieved from various addresses.
[1g]	wherein terminals adapted to act as	OCAs can also download updates from each other – minimizing significant usage of internet “backbone” capacity during the update cycle. The OCAs work in a network to distribute updates

	relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server; and	among each other and to include further OCAs to which updates and content can be sent. *See* Open Connect Overview, p. 5; Fill Patterns, pp. 1-3. Netflix' OCA that are adapted to act as relay servers (see 1d above) are adapted to modify transport requests received from the main server or from other relay servers and transmit the modified transport request to selected target terminals that includes addresses of further target terminals. The CCS server will to order OCA terminals to peer or tier fill from using OCAs selected by the CCS server. The CCS server uses Appliance Section Criteria to select OCA terminals as targets in the to transfer titles from: 1) the terminal appliance that receives the most-specific route; 2) the terminal appliance that receives the route to the client's netblock with the shortest AS path; 3) the terminal appliance that receives the route to the client's netblock with the lowest multi-exit discriminator; 4) the geographically closest appliance. The CCS server includes the URL addresses of these master or target terminals in the desired manifest, which is loaded by an OCA terminal in its memory or hard drive space in order to select an OCA for downloading titles from. After receiving the desired manifest, an OCA terminal will compare the manifest with its own "actual manifest", or the list of titles currently residing with the OCA terminal. The titles missing from the actual manifest are termed the "delta" or difference between actual and desired manifest. The OCA terminal will then query the CCS terminal for a list of download locations for each title on the delta. The CCS responds, as stated in [1f] with a list of URLs that are downloadable locations of master OCAs for each individual title needed by an OCA to fill its delta: "OCAs communicate at regular intervals with the control plane services, requesting (among other things) a manifest file that contains the list of titles they should be storing and serving to members. If there is a **delta** between the list of titles in the manifest and what they are currently storing, each OCA will send a request, during its configured fill window, **that includes a list of the new or updated titles that it needs**. The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title." (Emphasis added.) *See* https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 The action of an OCA requesting download locations (master OCAs) for its delta list from the CCS, and then requesting a delta-listed title from the list of a master OCAs, is equivalent to a modified

		transport request. The OCA is using a modified list of titles (delta or missing titles list from the desired manifest) to request a title or titles from further target terminals, or master OCAs. This is equivalent to: “terminals adapted to act as relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server.” The function of the two actions are substantially the same, which is to transmit to other selected relay servers or OCAs a modification of the original transport request, or desired manifest, along with URLs of those relay servers or URLs of master OCAs. The way the actions are performed are substantially the same. A delta is a modification of the manifest list, or in other words a subset list of what it is supposed to download. Instead of the terminal, or OCA, transmitting the delta list to another OCA, an “actual manifest” is sent to the CCS server which responds to the OCA with a list of URL locations to download the modified list titles of its delta list, after which the delta list is then sent to another OCA via a series of modified transfer requests. The result of these actions are substantially similar: further relay terminal addresses are sent to the OCAs, a modification of the original transport request (desired manifest) is transmitted to selected relay terminals (single or multiple delta titles to selected OCAs) are sent to addresses (URLs) of further selected relay servers (OCAs) in the form of a request for one or more titles from one or more master OCAs. Alternatively, and equivalently, Netflix documentation discloses that OCA terminals, if they are clustered or if they are in the same subnet, will attempt to peer or tier fill from each other. https://openconnect.zendesk.com/hc/en-us/articles/360035618071-Fill-patterns OCA terminals in a subnet or cluster broadcast their IP and physical locations to one another and save this information. In general, appliances determine where to receive fill using selection criteria that is used by Netflix client devices. The OCA terminals then use a similar Appliance Section Criteria as the CCS server uses to select OCA terminals as targets in the subnet or cluster to transfer titles from: 1) the terminal appliance that receives the most-specific route; 2) the appliance terminal that receives the route to the client’s netblock with the shortest AS path; 3) the appliance terminal that receives the route to the client’s netblock with the lowest multi-exit discriminator; 4) the geographically closest terminal. The OCA terminal includes the URL addresses of these terminals

		in its memory or hard drive space in order to select an OCA for downloading titles from. See *Fill Patterns*, pp. 1-4. After receiving the desired manifest, an OCA terminal will compare the manifest with its own "actual manifest", or the list of titles currently residing with the OCA terminal. The titles missing from the actual manifest are termed the "delta" or difference between actual and desired manifest. After selecting an OCA master using the selection criteria, the OCA terminal will transmit and request the delta list items to the selected OCA(s) in the subnet or cluster in the form of download requests for each title using the URL of the target OCA. This is equivalent to: "terminals adapted to act as relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server." The function of the two actions are substantially the same, which is to transmit to other selected relay servers or OCAs a modified list of the original transport request or manifest. The way the actions are performed are substantially the same. A modified desired manifest list is a subset list called the "delta." A relay server (OCA) creates a modified transport request (delta list) of its missing titles and transports this list to another relay server (OCA sends requests to other selected OCAs for the titles on the delta list in a series of requests) and includes addresses of the other relay servers (URLs of the selected OCA terminals in the request as an address to the selected OCAs). The result of these actions are substantially similar: a modification of the original transport request (desired manifest) is transmitted to selected relay terminals (selected OCAs) with addresses (URLs) of the selected relay servers (OCAs).
[1h]	wherein data to be retrieved by said target terminals are divided into a series of packets for transmission to said target	Netflix' OCAs are adapted to communicate with the main server which is hosted by AWS. The OCA's "Report their status to the Open Connect control panel services in Amazon Web Services": The Open Connect network can make parallelized cache fill transfers. As such, it must be transferring "packets" of the file. There is evidence that each file is downloaded for cache fill in parallel in packet sizes up to 16 kilobytes.

	terminals and each of said terminals is adapted to communicate directly with said main server to acknowledge receipt of the last packet of a series routed thereto.	Additionally, the last packet of a download for a cache fill is associated with a notice to the CCS that the download has been completed. The system can also use parallel processes to perform cache transfer using TCP/IP protocols. This is only possible if the CCS is working on a different section of the file at the same time, which is packetized file transfer. There is also evidence that the cache fill transfer uses TCP/IP protocols for packetized data transfer. TCP data communication protocol, which most of the Internet included Netflix uses, requires an acknowledgement of packets when content is transmitted: TCP is a reliable byte stream delivery service which guarantees that all bytes received will be identical and in the same order as those sent. Since packet transfer by many networks is not reliable, **TCP achieves this using a technique known as positive acknowledgement with re-transmission**. This **requires the receiver to respond with an acknowledgement message as it receives the data**. The sender keeps a record of each packet it sends and maintains a timer from when the packet was sent. The sender re-transmits a packet if the timer expires before receiving the acknowledgement. The timer is needed in case a packet gets lost or corrupted.[Comer, Douglas E. (2006). Internetworking with TCP/IP: Principles, Protocols, and Architecture. Vol. 1 (5th ed.). Prentice Hall. ISBN 978-0-13-187671-2 Netflix requires network traffic to OCA be in TCP protocol:  • **Traffic from OCA**: Allow all destination addresses and ports. • **Traffic to OCA**: Allow TCP 22, 53, 80, 179, 443, UDP 53 and 123 (source and destination), ICMP types 0, 3, 8, 11, and all ICMPv6 from any public IP/port. Allow all return traffic from any appliance-initiated connection (TCP established). https://openconnect.zendesk.com/hc/en-us/articles/360035533071#routi



Glossary of terms

CAGR
Compound Annual Growth Rate

Cache
A temporary local copy of information that originated elsewhere. Thus for CDNs, a copy of files to be delivered to consumers, stored in a local server

CDN (Content Delivery Network)
A distributed system of servers, designed to enable the efficient and reliable distribution of content over the internet

CP (content provider)
An internet business whose focus is delivering content (rather than - say - e-commerce) to consumers. Netflix, CNN and YouTube are examples

FBB
Fixed broadband

Hop
One step in a packet of data's journey across multiple servers

ISP (Internet Service Provider)
A company providing internet connectivity to consumers (consumers or businesses). May provide fixed broadband, mobile data or both

IXP (Internet Exchange Point)
A location where many networks meet to exchange traffic, avoiding the need for multiple bilateral connections

Latency
The lag between a packet of data being sent and reaching its destination

OTT (Over The Top)
Describes services delivered over another network without being integrated with it. YouTube, Facebook and Netflix are examples (since they are not provided by telcos operating broadband networks)

Packet loss
When a router is sent more data than it can handle, it discards a certain amount of data. This is known as packet loss. Typically the data in question will then be requested again from the source server

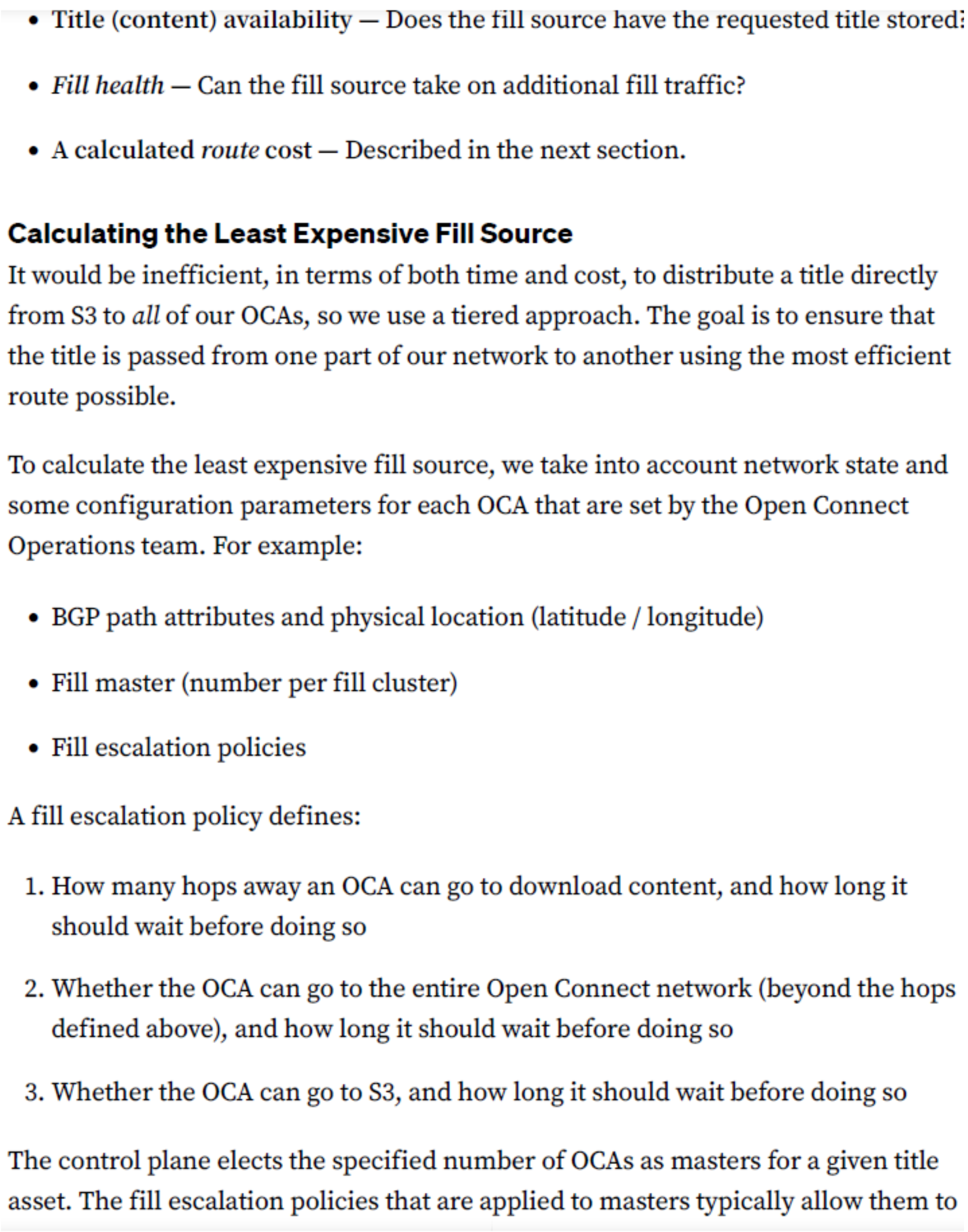
Router
A switch on the internet, that receives packets of data and sends them onwards down the appropriate link

Server
A computer that stores and transmits content

TCP/IP
Transmission control protocol and internet protocol. The two foundational standards for data transmission that underpin the internet

		"A cooperative approach to content delivery," Netflix (2021), 37. Where does Netflix use TCP? Netflix uses TCP for internet streaming to send packets of data for video. Additionally, Netflix specifically looks at the number of TCP connections to determine internet speeds in accordance with testing of OCAs. In another example, "After **the fill master OCA** has completed its S3 download, **it reports back to the control plane** that it now has the title stored." See https://netflixtechblog.com/netflix-and-fill-c43a32b490c0
[3a]	The network as claimed in claim 1,	*See* above.
[3b]	wherein terminals acting as relay servers are adapted to select further downstream target terminals to act as further relay servers on the basis of their relative performances of the further target terminals indicated in said transport request.	OCAs can also download updates from each other – minimizing significant usage of internet "backbone" capacity during the update cycle. The OCAs work in a network to distribute updates among each other and to include further OCAs to which updates and content can be sent. *See* Open Connect Overview, p. 5. Alternatively, and equivalently, Netflix documentation discloses that OCA terminals, if they are clustered or if they are in the same subnet, will attempt to peer or tier fill from each other. https://openconnect.zendesk.com/hc/en-us/articles/360035618071-Fill-patterns OCA terminals in a subnet or cluster broadcast their IP and physical locations to one another and save this information. In general, appliances determine where to receive fill using selection criteria that is used by Netflix client devices. The OCA terminals then use a similar Appliance Section Criteria as the CCS server uses to select OCA terminals as targets in the subnet or cluster to transfer titles from: 1) the terminal appliance that receives the most-specific route; 2) the appliance terminal that receives the route to the client's netblock with the shortest AS path; 3) the appliance terminal that receives the route to the client's netblock with the lowest multi-exit discriminator; 4) the

		geographically closest terminal. The OCA terminal includes the URL addresses of these terminals in its memory or hard drive space in order to select an OCA for downloading titles from. See *Fill Patterns*, pp. 1-4. After receiving the desired manifest, an OCA terminal will compare the manifest with its own “actual manifest”, or the list of titles currently residing with the OCA terminal. The titles missing from the actual manifest are termed the “delta” or difference between actual and desired manifest. After selecting an OCA master using the selection criteria, the OCA terminal will transmit and request the delta list items to the selected OCA(s) in the subnet or cluster in the form of download requests for each title using the URL of the target OCA. .



- Title (content) availability — Does the fill source have the requested title stored?
- *Fill health* — Can the fill source take on additional fill traffic?
- A calculated *route* cost — Described in the next section.

Calculating the Least Expensive Fill Source

It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to *all* of our OCAs, so we use a tiered approach. The goal is to ensure that the title is passed from one part of our network to another using the most efficient route possible.

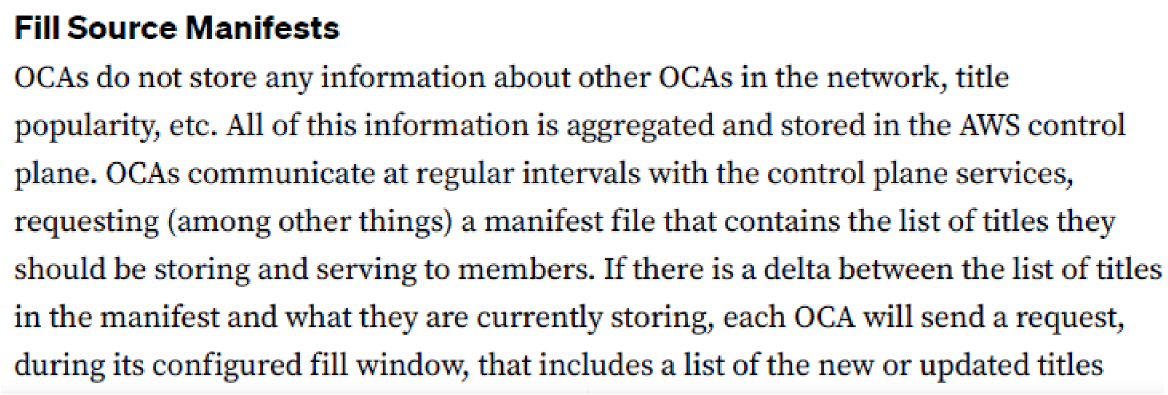
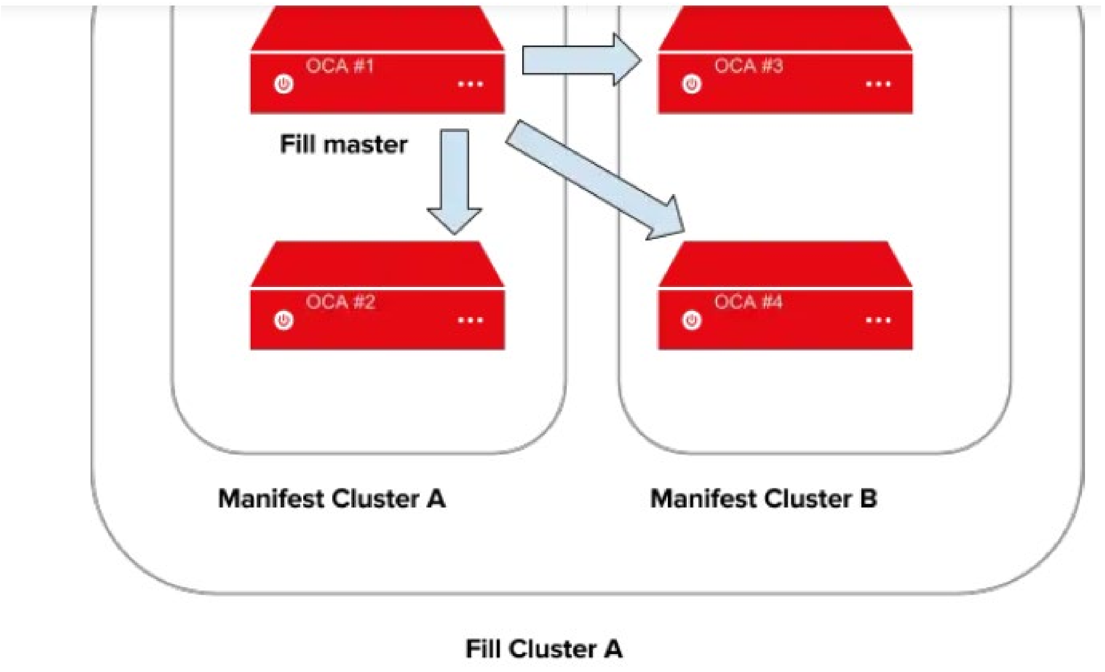
To calculate the least expensive fill source, we take into account network state and some configuration parameters for each OCA that are set by the Open Connect Operations team. For example:

- BGP path attributes and physical location (latitude / longitude)
- Fill master (number per fill cluster)
- Fill escalation policies

A fill escalation policy defines:

1. How many hops away an OCA can go to download content, and how long it should wait before doing so
2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so
3. Whether the OCA can go to S3, and how long it should wait before doing so

The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to

		 **Fill Source Manifests** OCAs do not store any information about other OCAs in the network, title popularity, etc. All of this information is aggregated and stored in the AWS control plane. OCAs communicate at regular intervals with the control plane services, requesting (among other things) a manifest file that contains the list of titles they should be storing and serving to members. If there is a delta between the list of titles in the manifest and what they are currently storing, each OCA will send a request, during its configured fill window, that includes a list of the new or updated titles  OCA #1 OCA #3 Fill master OCA #2 OCA #4 Manifest Cluster A Manifest Cluster B Fill Cluster A

		When the second tier of OCAs complete their download, they report back their status, other OCAs can then fill from them, and so on. This process continues during the fill window. If there are titles being stored on an OCA that are no longer needed, they are put into a delete manifest and then deleted after a period of time that ensures we don't interrupt any live sessions. As the sun moves west and more members begin streaming, the fill window in this time zone ends, and the fill pattern continues as the fill window moves across other time zones — until enough of the OCAs in our global network that need to be able to serve this new title have it stored.

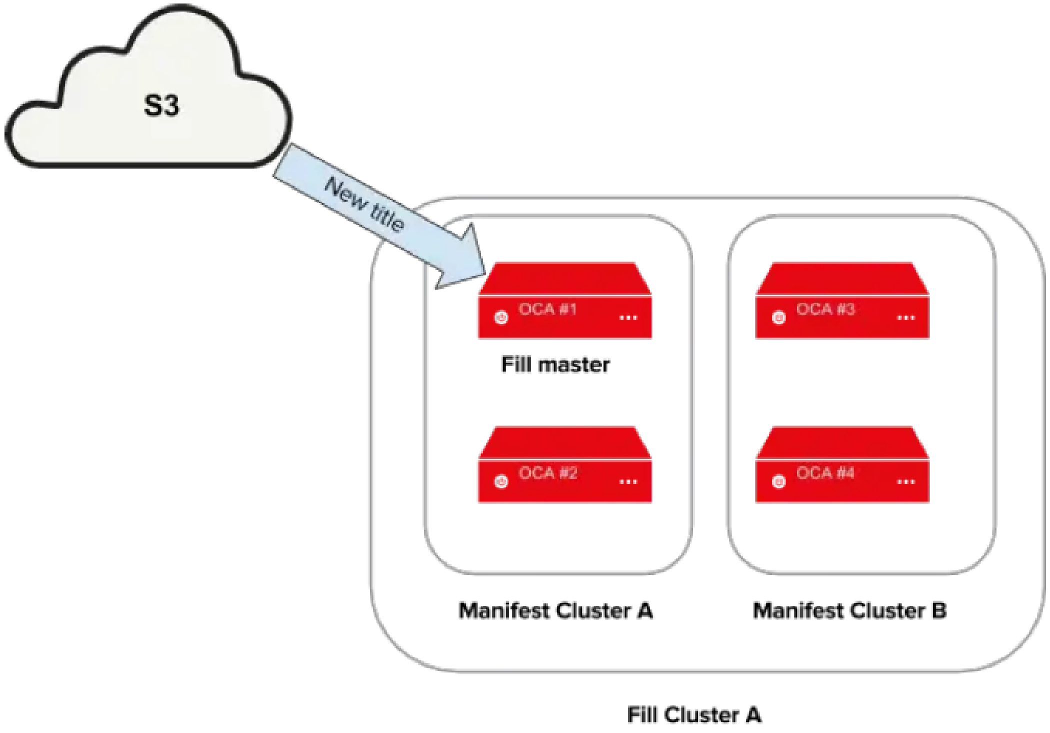
1. Peer fill: Available OCAs within the same manifest cluster or the same subnet

2. Tier fill: Available OCAs outside the manifest cluster configuration

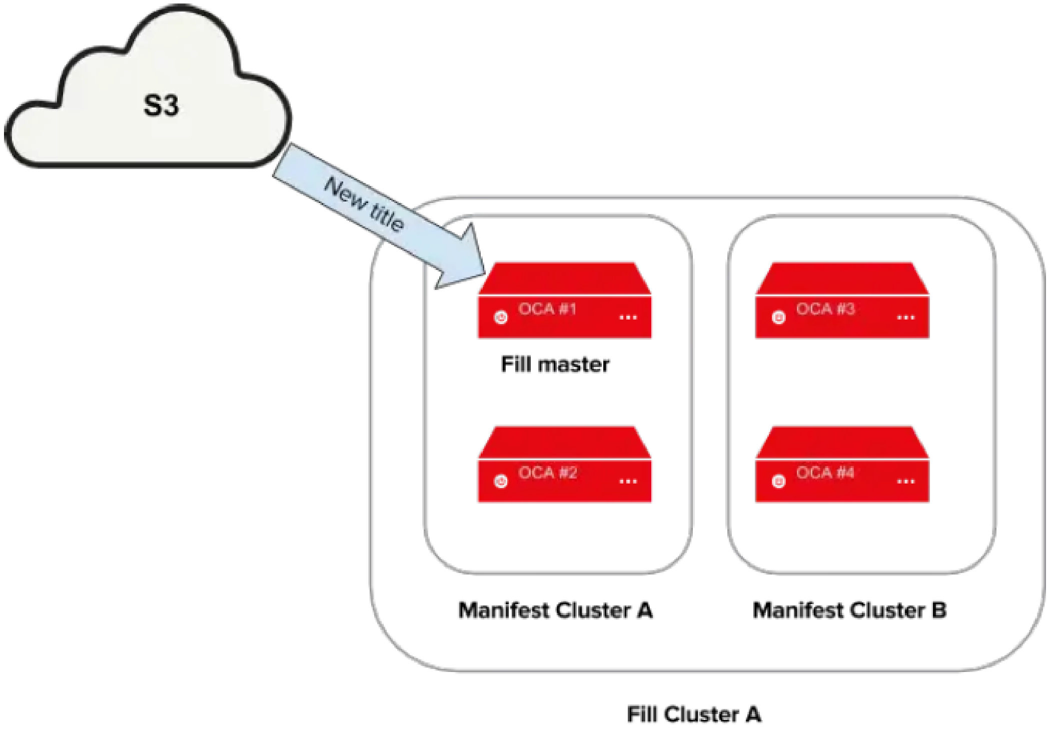
3. Cache fill: Direct download from S3

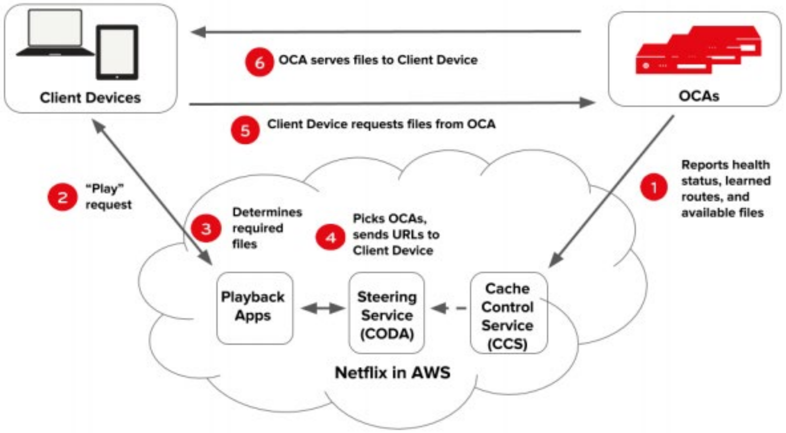
Example Scenario

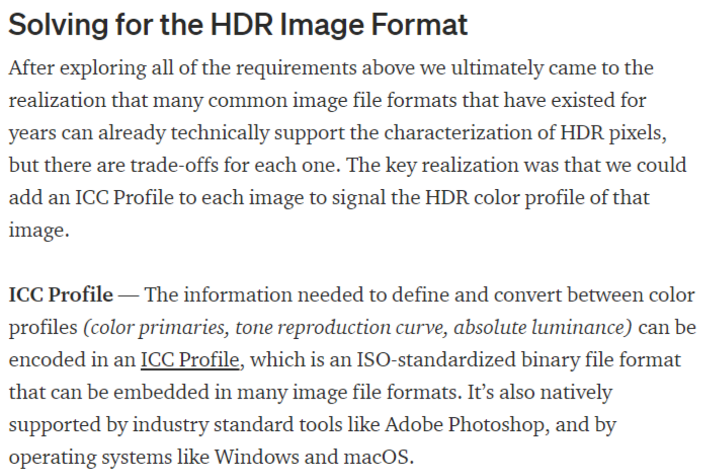
After the fill master OCA has completed its S3 download, it reports back to the control plane that it now has the title stored. The next time the other OCAs communicate with the control plane to request a fill source for this title, they are given the option to fill from the fill master.

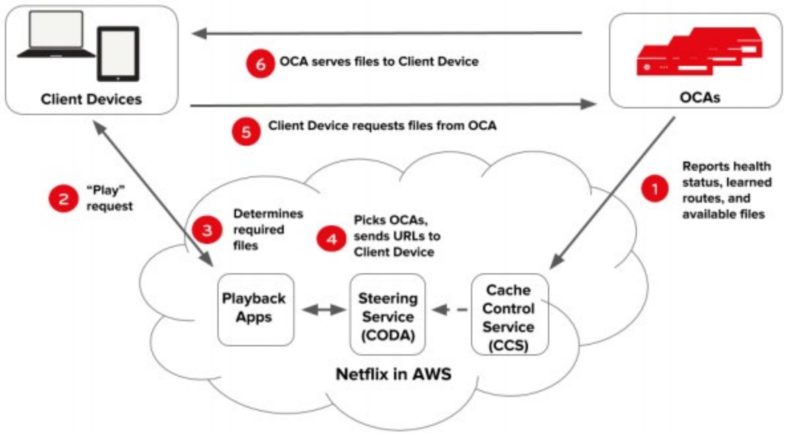


		The control panel on the main server observes performance data to determine which “second tier” OCAs to complete their downloads with from a master OCA, to “report back their (second tier OCAs) status” to the main server, “and so on” to send transport requests to each new “master” OCA and new “second tier” OCA’s, “and so on.”
[4a]	The network as claimed in claim 1,	*See* above.
[4b]	wherein the first server is a terminal adapted to act as relay server.	In the Netflix architecture diagram, a terminal can act as the first server, where the other OCAs can have the option of downloading from the “fill master OCA.”

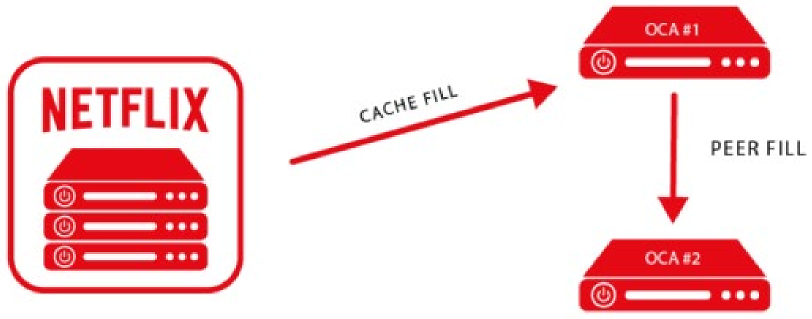
		1. Peer fill: Available OCAs within the same manifest cluster or the same subnet 2. Tier fill: Available OCAs outside the manifest cluster configuration 3. Cache fill: Direct download from S3 **Example Scenario** After the fill master OCA has completed its S3 download, it reports back to the control plane that it now has the title stored. The next time the other OCAs communicate with the control plane to request a fill source for this title, they are given the option to fill from the fill master.  S3 New title OCA #1 Fill master OCA #2 OCA #3 OCA #4 Manifest Cluster A Manifest Cluster B Fill Cluster A

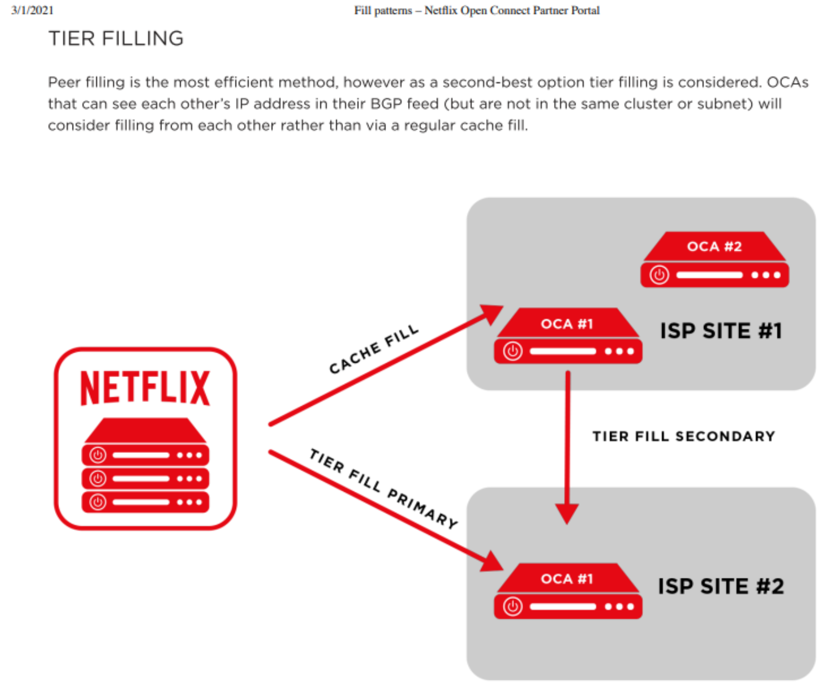
[5a]	The network as claimed in claim 1,	*See* above.
[5b]	wherein each of said terminals is adapted to communicate directly with said main server in an upstream direction.	Each of the OCAs can report directly to the Netflix amin server in an upstream direction.  Client Devices OCAs 6 OCA serves files to Client Device 5 Client Device requests files from OCA 1 Reports health status, learned routes, and available files 2 "Play" request 3 Determines required files 4 Picks OCAs, sends URLs to Client Device Playback Apps Steering Service (CODA) Cache Control Service (CCS) Netflix in AWS
[6a]	The network as claimed in claim 1,	*See* above.
[6b]	wherein data is routed to said terminals as routed network protocol traffic such as TCP/IP traffic.	Netflix uses TCP for internet streaming to send packets of data for video. Additionally, Netflix specifically looks at the number of TCP connections to determine internet speeds in accordance with testing of OCAs. See [1h] above.
[8a]	The network as claimed in claim 1,	*See* above.

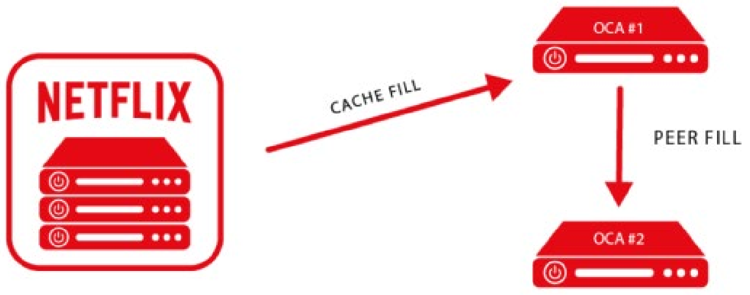
[8b]	wherein data is transmitted in binary format.	Netflix states, on its public technical blog, that binary format data can be used in image file formats sent through the network. *See* https://netflixtechblog.com/enhancing-the-netflix-ui-experience-with-hdr-1e7506ad3e8  **Solving for the HDR Image Format** After exploring all of the requirements above we ultimately came to the realization that many common image file formats that have existed for years can already technically support the characterization of HDR pixels, but there are trade-offs for each one. The key realization was that we could add an ICC Profile to each image to signal the HDR color profile of that image. **ICC Profile** — The information needed to define and convert between color profiles *(color primaries, tone reproduction curve, absolute luminance)* can be encoded in an ICC Profile, which is an ISO-standardized binary file format that can be embedded in many image file formats. It's also natively supported by industry standard tools like Adobe Photoshop, and by operating systems like Windows and macOS.
[9a]	A method of operating a data communication network, the data communication network comprising: a plurality of terminals	Netflix uses a method to deliver Netflix TV shows and movies to members using world-wide system called **Open Connect**. The building blocks of Open Connect are our suite of purpose-built **server appliances, called Open Connect Appliances (OCAs)**. *See* Open Connect Overview, p. 2. These are deployed directly inside ISP networks. Netflix provides the server hardware. The OCAs report to a Open Connect control plane to control fill behavior (adding new files to OCAs nightly) and to compute and/or store data. *See id.* p. 3-4. Accordingly, OCAs include both an input mechanism and display mechanism.

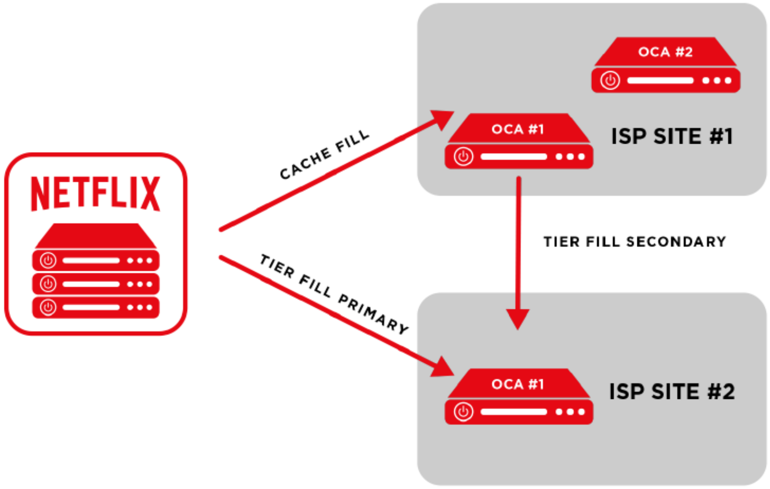
		 Client Devices OCAs 6 OCA serves files to Client Device 5 Client Device requests files from OCA 2 "Play" request 1 Reports health status, learned routes, and available files 3 Determines required files 4 Picks OCAs, sends URLs to Client Device Playback Apps Steering Service (CODA) Cache Control Service (CCS) Netflix in AWS Open Connect Appliances can be embedded in your ISP network. Embedded OCAs have the same capabilities as the OCAs that we use in our 60+ global data centers, and they are provided to qualifying ISP partners at no charge. Each embedded OCA deployment will offload a substantial amount of Netflix content traffic from peering or transport circuits. Multiple physical deployments can be distributed or clustered on a geographic or network basis to maximize local offload. Source: https://openconnect.netflix.com/en/#sample-architectures
[9b]	a network information database and a main server adapted to manage selective retrieval of data from a first server by at least one target terminal selected from said plurality of	Netflix runs the operation of Open Connect from a Netflix application (CSS server) that is hosted in AWS. *See* Open Connect Overview, p. 4-5. In its global network, Netflix provides **data centers** such as an "S3" server **("first server")** housing **content ("data") on at lease one server**, and provides OCA users (such as ISP's) direct access to these data centers over the Internet that are housing the content. One or more of these **data centers** house a "**first server**" according to the claims. In deployment of Open Connect, Netflix provides Internet Service Providers with an OCA appliance direct "settlement-free interconnection (SFI)." The terminal OCA can "Connect via direct Private Network Interconnect (PNI) or IXP-based SFI peering to Netflix Open Connect Appliances **in our data centers**."

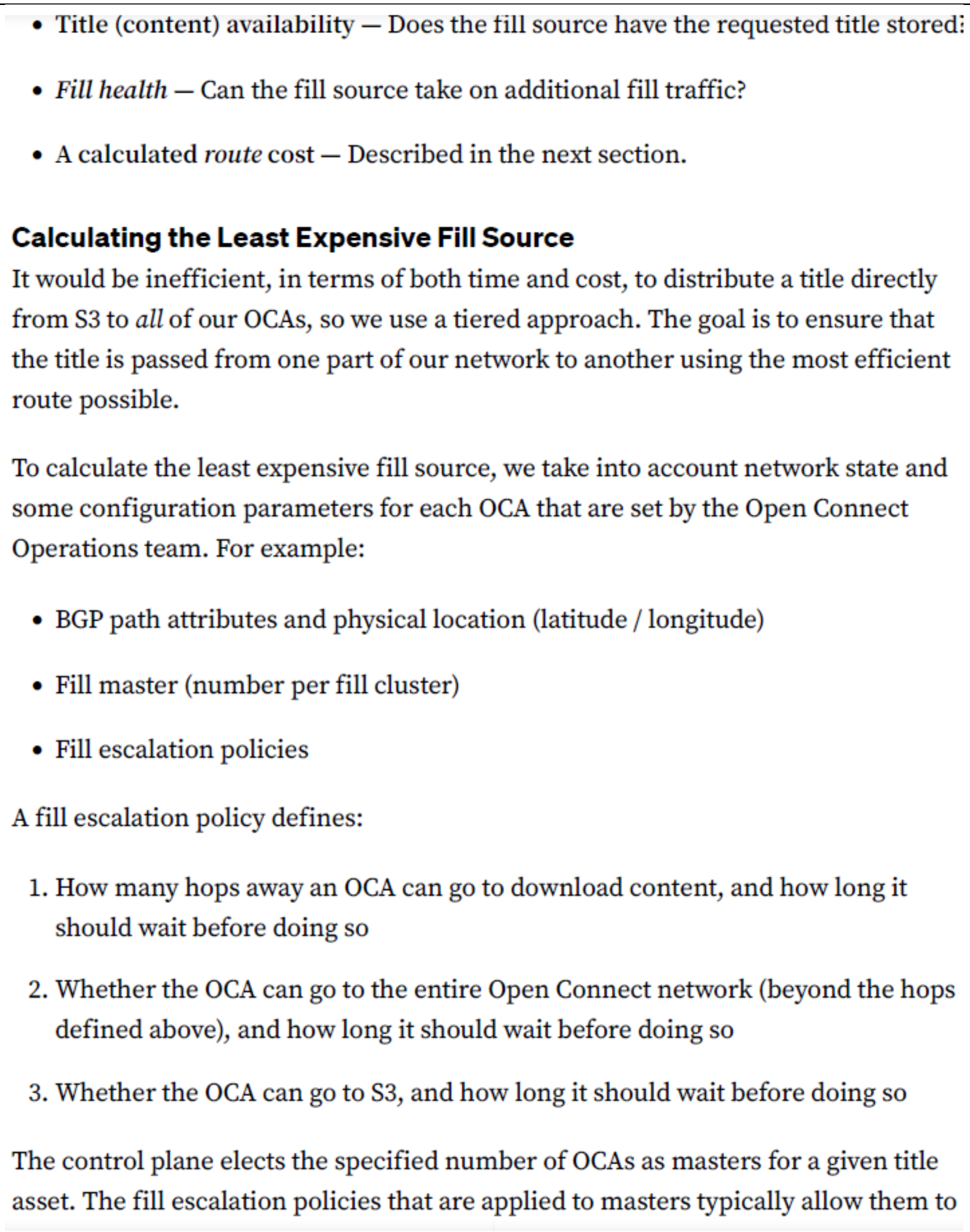
	terminals; comprising	“Netflix has the ability to interconnect at a number of global data center facilities and public Internet Exchange fabrics as listed on our Peering Locations page. We openly peer with any network at IXP locations where we are mutually present and we consider private interconnection as appropriate.” ISPs who do not currently participate in pubic peering might want to consider that a single IX port can support multiple peering sessions, providing **direct access to various content,** cloud, and network providers. Welcome to Open Connect, p. 3. Dkt 39 at p. 48. The following diagram also illustrates access from a target terminal (OCA #1, OCA #2) to a Netflix first server in “our data centers.” OCAs in a cluster and on the same subnet can attempt peer filling from each other. There is also Tier filling where if in different ISP sites. deploymentguide.pdf (netflix.com)

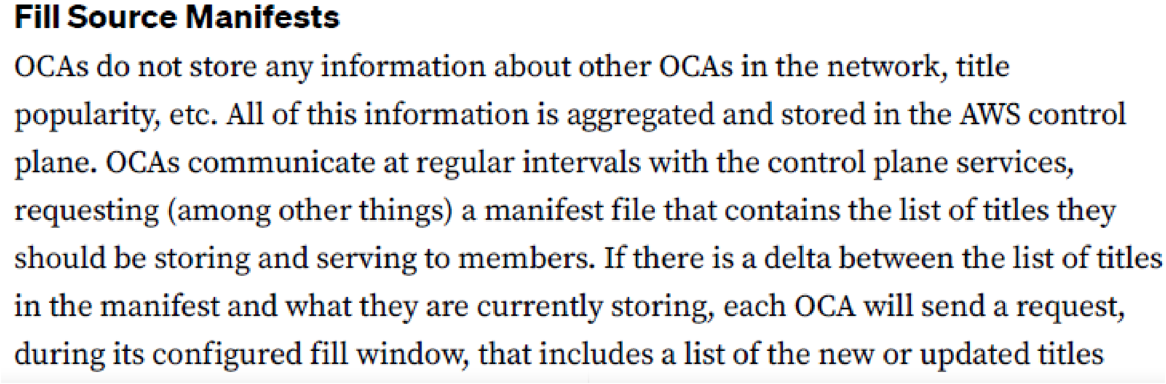
		 https://openconnect.zendesk.com/hc/en-us/articles/360035618071 2/4

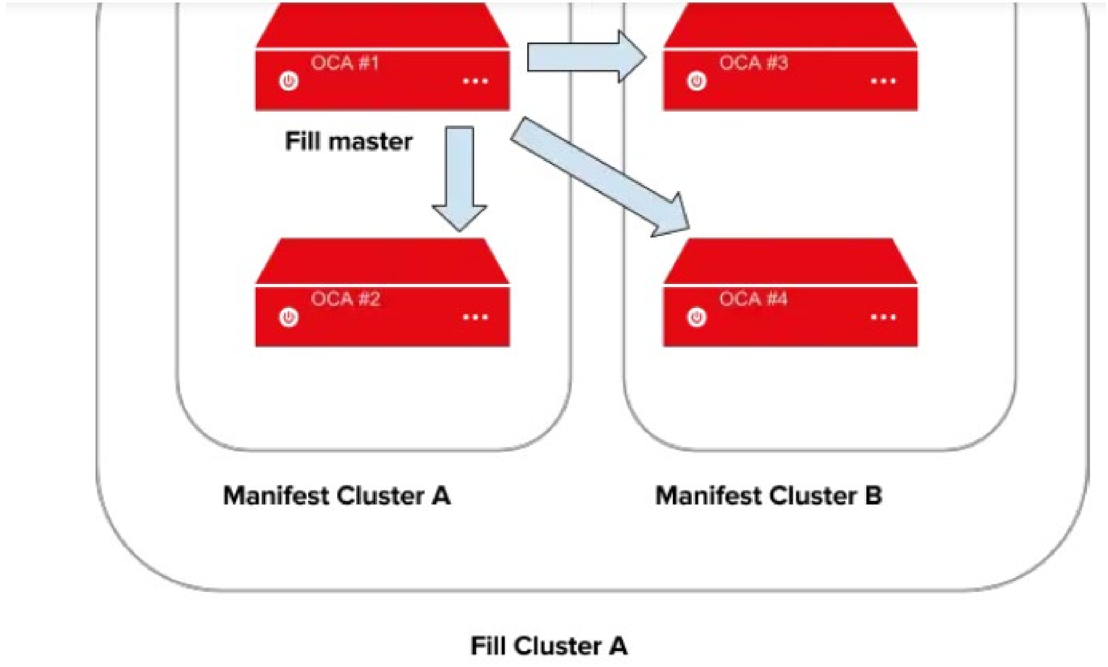
		 3/1/2021 Fill patterns – Netflix Open Connect Partner Portal TIER FILLING Peer filling is the most efficient method, however as a second-best option tier filling is considered. OCAs that can see each other's IP address in their BGP feed (but are not in the same cluster or subnet) will consider filling from each other rather than via a regular cache fill. NETFLIX CACHE FILL TIER FILL PRIMARY TIER FILL SECONDARY OCA #2 OCA #1 ISP SITE #1 OCA #1 ISP SITE #2 The "target terminal selected form said plurality of terminals" language of the claim is infringed by the health and performance monitoring and OCA target terminal selection process described below in Sec. [1e].
[9c]	operating at least two of said terminals as relay servers for serving data retrieved from said first server to at least one target terminal, wherein said main server is	A "first server" or S3 is identified above in Sec. [1a] as a server within Netflix' data centers , and at least two of the OCAs are adapted to act as relay servers for serving data retrieved from said first server to at least one OCA terminal. According to Netflix' network architecture, OCAs in a cluster and on the same subnet can attempt peer filling content "cache fill" from the first server (data center) to OCA #1. OCA#1 then acts as a "relay server" to fill cached content to each other, from OCA#1 to OCA#2. OCA#2 becomes a

	distinct from said first server, and further comprising:	“target terminal” in this example that is served data from the relay server (OCA#1) retrieved from the first server (data center). See https://openconnect.netflix.com/deploymentguide.pdf.  https://openconnect.zendesk.com/hc/en-us/articles/360035618071 2/4 In another example, Netflix’s first server (data center) can cache fill content to a terminal OCA#1, which then acts as a “relay server” to fill a target terminal (OCA#2) using “tier filling.” “Peer filling is the most efficient method, however as a second-best option tier filling is considered. OCAs that can see each other’s IP address in their BGP feed (but are not in the same cluster or subnet) will consider filling from each other rather than via a regular cache fill.”

		 NETFLIX CACHE FILL TIER FILL PRIMARY OCA #2 OCA #1 ISP SITE #1 TIER FILL SECONDARY OCA #1 ISP SITE #2 See https://openconnect.netflix.com/deploymentguide.pdf. In another example, https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states that there are “**The control plane elects the specified number of OCAs as masters. . . .**” OCAs can act as relay servers, or “masters” that target terminals can use to gain, or fill, content:

		 • Title (content) availability — Does the fill source have the requested title stored? • *Fill health* — Can the fill source take on additional fill traffic? • A calculated *route* cost — Described in the next section. **Calculating the Least Expensive Fill Source** It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to *all* of our OCAs, so we use a tiered approach. The goal is to ensure that the title is passed from one part of our network to another using the most efficient route possible. To calculate the least expensive fill source, we take into account network state and some configuration parameters for each OCA that are set by the Open Connect Operations team. For example: • BGP path attributes and physical location (latitude / longitude) • Fill master (number per fill cluster) • Fill escalation policies A fill escalation policy defines: 1. How many hops away an OCA can go to download content, and how long it should wait before doing so 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so 3. Whether the OCA can go to S3, and how long it should wait before doing so The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to

		 **Fill Source Manifests** OCAs do not store any information about other OCAs in the network, title popularity, etc. All of this information is aggregated and stored in the AWS control plane. OCAs communicate at regular intervals with the control plane services, requesting (among other things) a manifest file that contains the list of titles they should be storing and serving to members. If there is a delta between the list of titles in the manifest and what they are currently storing, each OCA will send a request, during its configured fill window, that includes a list of the new or updated titles

		OCA #1 OCA #3 Fill master OCA #2 OCA #4 Manifest Cluster A Manifest Cluster B Fill Cluster A

		When the second tier of OCAs complete their download, they report back their status, other OCAs can then fill from them, and so on. This process continues during the fill window. If there are titles being stored on an OCA that are no longer needed, they are put into a delete manifest and then deleted after a period of time that ensures we don't interrupt any live sessions. As the sun moves west and more members begin streaming, the fill window in this time zone ends, and the fill pattern continues as the fill window moves across other time zones — until enough of the OCAs in our global network that need to be able to serve this new title have it stored.

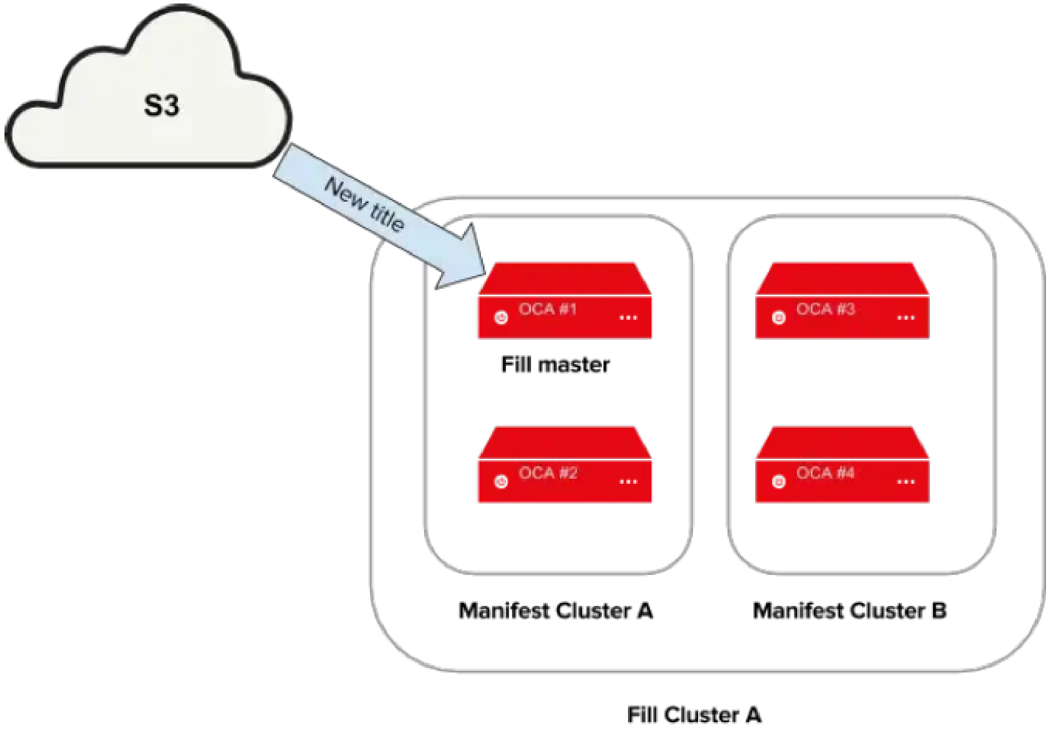
1. Peer fill: Available OCAs within the same manifest cluster or the same subnet

2. Tier fill: Available OCAs outside the manifest cluster configuration

3. Cache fill: Direct download from S3

Example Scenario

After the fill master OCA has completed its S3 download, it reports back to the control plane that it now has the title stored. The next time the other OCAs communicate with the control plane to request a fill source for this title, they are given the option to fill from the fill master.



[9d]	sending transport requests from the main server to at least one first target terminal based on terminal performance information stored in the network information database; and operating the first target terminal to act as relay server;	All OCA deployments are constantly monitored to ensure reliability and efficiency. Netflix **makes use of non-peak bandwidth to download the vast majority of content updates to the OCAs in network during these configurable time windows. OCAs can also download updates from each other – minimizing significant usage of internet “backbone” capacity during the update cycle.** A “desired manifest” and when needed an “emergency manifest” are posted by Netflix to the CCS server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. A “desired manifest” and when needed an “emergency manifest” are posted by Netflix to the CCS (Control Content Service) server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. Equivalently, these manifests are fetched by regularly from the CCS by each OCA terminal, which traverses the desired manifest or emergency manifest. The CCS server acts as an email inbox for manifests sent directly to the OCAs, where each OCA terminal is mandated to check regularly for manifests and download the manifests to the local OCA terminal hard drive. This desired manifest and emergency manifest and the download location data on the CCS are equivalent the “server is adapted to send transport requests direct to at least one first target terminal.” The functions are substantially the same - to send a request from a main server (CCS) to a terminal (OCA) with instructions to download or “fill” content. The way this is accomplished is substantially the same. The claim states that the main server is adapted to send transport requests direct to at least one first target terminal over a network, and on the Open Connect Network, Netflix posts a desired manifest or emergency manifest to the CCS server that is intended for each OCA in the network. The results are substantially the same - to transmit commands from the main server/CCS server to a terminal/OCA to download content data in the manifest that needs to be retrieved.

		The control plane elects the specified number of OCAs as masters for a given title asset. https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 Thus, the CCS server selects OCA terminals as a download location for a given title asset. The main server on CCS is further adapted to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis of their relative response times. At https://netflixtechblog.com/netflix-and-fill-c43a32b490c0, Netflix wrote: The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title. The determination of the list takes into consideration several high-level factors: Title (content) availability — Does the fill source have the requested title stored? **Fill health — Can the fill source take on additional fill traffic?** A calculated route cost — Described in the next section. A fill escalation policy defines: 7. How many hops away an OCA can go to download content, and **how long it should wait before doing so** 8. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), **and how long it should wait before doing so** 9. Whether the OCA can go to S3, **and how long it should wait before doing so** (Emphasis added.) The CCS server monitors the "fill health" of each potential download locations, or fill sources that includes OCAs, and uses a fill escalation policy based upon response times to determine if the performance of the download location can take on additional traffic or not and how long the OCA should wait for a response before escalation to the next download locations. Thus, the manifest is assembled by the CCS server and download locations are assembled based on OCA performance

		information, and the manifest is sent to a given OCA. Therefore, the transport request is sent on a basis of said terminal performance information.
[9e]	operating the main server to monitor the response times of terminals in the network and selecting terminals to act as relay servers for particular data transfer on the basis of their relative response times;	All OCA deployments are constantly monitored to ensure reliability and efficiency. Netflix **makes use of non-peak bandwidth to download the vast majority of content updates to the OCAs in network during these configurable time windows. OCAs can also download updates from each other – minimizing significant usage of internet "backbone" capacity during the update cycle.** The main server on CCS is further adapted to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis of their relative response times. At https://netflixtechblog.com/netflix-and-fill-c43a32b490c0, Netflix wrote: The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title. The determination of the list takes into consideration several high-level factors: Title (content) availability — Does the fill source have the requested title stored? **Fill health — Can the fill source take on additional fill traffic?** A calculated route cost — Described in the next section. A fill escalation policy defines: 1. How many hops away an OCA can go to download content, and **how long it should wait before doing so** 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), **and how long it should wait before doing so** 3. Whether the OCA can go to S3, **and how long it should wait before doing so** (Emphasis added.)

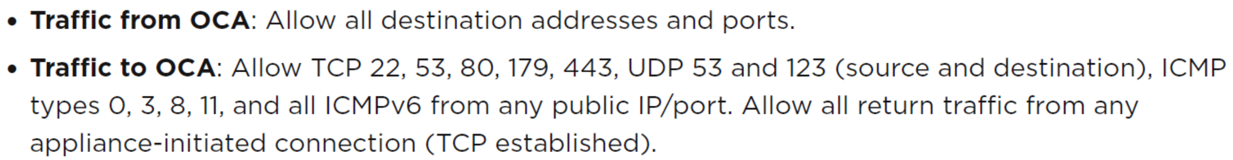
		The CCS server monitors the “fill health” of each potential download locations, or fill sources that includes OCAs, and uses a fill escalation policy based upon response times to determine if the performance of the download location can take on additional traffic or not and how long the OCA should wait for a response before escalation to the next download locations.
[9f]	wherein each such transport request includes details of data to be retrieved, the address of the first server from which the data is to be requested by the first target terminal, addresses of at least one second target terminal to which the data retrieved from the first server is to be relayed by the first target terminal and an indication of a relative performance of a further target terminal based on the terminal performance information stored	A “desired manifest” and when needed an “emergency manifest” are transport requests posted by Netflix to the CCS (Control Content Service) server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. these manifests are fetched by regularly from the CCS by each OCA terminal, which traverses the desired manifest or emergency manifest. OCAs then then query the CCS for location information files that list where each title on the desired manifest that is needed by an OCA can be downloaded. https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states a master OCA and at least one second OCA can be selected based on their relative performance: The CCS server information location list provides the address of a first server, called the “S3” server, for a download location and provides other download locations of a second and additional master OCAs: It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to all of our OCAs, so we use a tiered approach. . . . A fill escalation policy defines: 1. How many hops away an OCA can go to download content, and how long it should wait before doing so 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so 3. Whether the OCA can go to S3, and how long it should wait before doing so

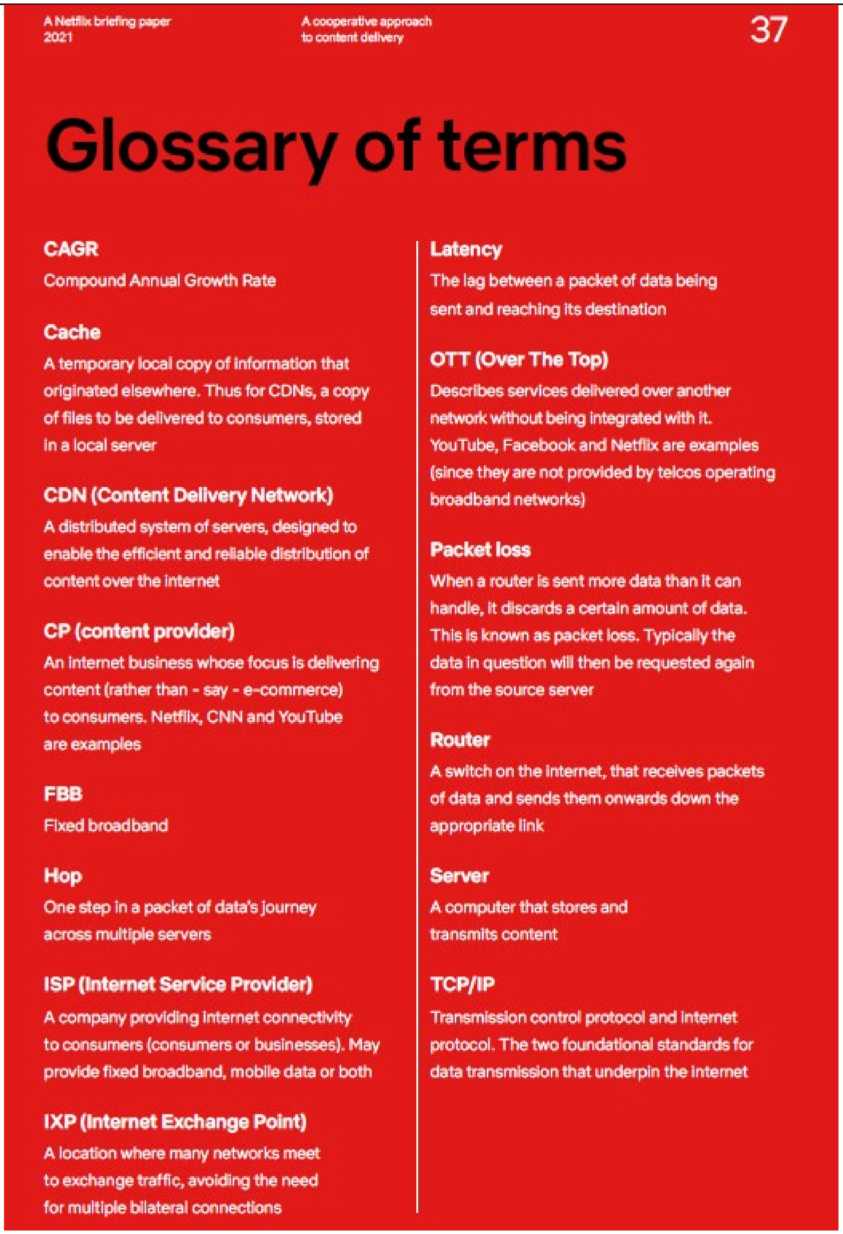
	in the network information database;	The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to reach farther with less delay in order to grab that content and then share it locally with non-masters. As stated in [1e], the CCS monitors the "fill health" and performance of download locations, which is based on performance of the OCAs, to determine if that OCA will be selected as a download location or not. This desired manifest and emergency manifest along with the downloaded location information files and fill policy for master OCAs and a second (target terminal) OCA to fill from a (first terminal) OCA on the CCS are equivalent the "each such transport request includes details of data to be retrieved, the address of the first server from which the data is to be requested by the first target terminal, the addresses of at least one second target terminal to which the data from the first server to be relayed by the first target terminal." The functions are substantially the same - to send a request from a main server (CCS) to a terminal (OCA) with instructions to download or "fill" content from specific master terminal/OCA addresses and includes an address of at least one second terminal/OCA. The way this is accomplished is substantially the same. The claim states that the main server is adapted to send transport requests that include download locations direct to at least one first target terminal over a network, and on the Open Connect Network, Netflix posts a desired manifest or emergency manifest to the CCS server along with download locations that are intended for each OCA to read on a regular basis. The results are substantially the same - to transmit commands from the main server/CCS server to a terminal/OCA to download content data in the manifest that needs to be retrieved from various addresses.
[9i]	operating terminals adapted to act as relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the	OCAs can also download updates from each other – minimizing significant usage of internet "backbone" capacity during the update cycle. The OCAs work in a network to distribute updates among each other and to include further OCAs to which updates and content can be sent. *See* Open Connect Overview, p. 5; Fill Patterns, pp. 1-3. Netflix' OCA that are adapted to act as relay servers (see 1d above) are adapted to modify transport requests received from the main server or from other relay servers and transmit the modified transport request to selected target terminals that includes addresses of further target terminals.

	modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server; and	The CCS server will to order OCA terminals to peer or tier fill from using OCAs selected by the CCS server. The CCS server uses Appliance Section Criteria to select OCA terminals as targets in the to transfer titles from: 1) the terminal appliance that receives the most-specific route; 2) the terminal appliance that receives the route to the client's netblock with the shortest AS path; 3) the terminal appliance that receives the route to the client's netblock with the lowest multi-exit discriminator; 4) the geographically closest appliance. The CCS server includes the URL addresses of these master or target terminals in the desired manifest, which is loaded by an OCA terminal in its memory or hard drive space in order to select an OCA for downloading titles from. After receiving the desired manifest, an OCA terminal will compare the manifest with its own "actual manifest", or the list of titles currently residing with the OCA terminal. The titles missing from the actual manifest are termed the "delta" or difference between actual and desired manifest. The OCA terminal will then query the CCS terminal for a list of download locations for each title on the delta. The CCS responds, as stated in [1f] with a list of URLs that are downloadable locations of master OCAs for each individual title needed by an OCA to fill its delta: "OCAs communicate at regular intervals with the control plane services, requesting (among other things) a manifest file that contains the list of titles they should be storing and serving to members. If there is a **delta** between the list of titles in the manifest and what they are currently storing, each OCA will send a request, during its configured fill window, **that includes a list of the new or updated titles that it needs**. The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title." (Emphasis added.) *See* https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 The action of an OCA requesting download locations (master OCAs) for its delta list from the CCS, and then requesting a delta-listed title from the list of a master OCAs, is equivalent to a modified transport request. The OCA is using a modified list of titles (delta or missing titles list from the desired manifest) to request a title or titles from further target terminals, or master OCAs. This is equivalent to: "terminals adapted to act as relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport

		request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server." The function of the two actions are substantially the same, which is to transmit to other selected relay servers or OCAs a modification of the original transport request, or desired manifest, along with URLs of those relay servers or URLs of master OCAs. The way the actions are performed are substantially the same. A delta is a modification of the manifest list, or in other words a subset list of what it is supposed to download. Instead of the terminal, or OCA, transmitting the delta list to another OCA, an "actual manifest" is sent to the CCS server which responds to the OCA with a list of URL locations to download the modified list titles of its delta list, after which the delta list is then sent to another OCA via a series of modified transfer requests. The result of these actions are substantially similar: further relay terminal addresses are sent to the OCAs, a modification of the original transport request (desired manifest) is transmitted to selected relay terminals (single or multiple delta titles to selected OCAs) are sent to addresses (URLs) of further selected relay servers (OCAs) in the form of a request for one or more titles from one or more master OCAs. Alternatively and equivalently, Netflix documentation discloses that OCA terminals, if they are clustered or if they are in the same subnet, will attempt to peer or tier fill from each other. https://openconnect.zendesk.com/hc/en-us/articles/360035618071-Fill-patterns OCA terminals in a subnet or cluster broadcast their IP and physical locations to one another and save this information. In general, appliances determine where to receive fill using selection criteria that is used by Netflix client devices. The OCA terminals then use a similar Appliance Section Criteria as the CCS server uses to select OCA terminals as targets in the subnet or cluster to transfer titles from: 1) the terminal appliance that receives the most-specific route; 2) the appliance terminal that receives the route to the client's netblock with the shortest AS path; 3) the appliance terminal that receives the route to the client's netblock with the lowest multi-exit discriminator; 4) the geographically closest terminal. The OCA terminal includes the URL addresses of these terminals in its memory or hard drive space in order to select an OCA for downloading titles from. See *Fill Patterns*, pp. 1-4. After receiving the desired manifest, an OCA terminal will compare the manifest with its own "actual manifest", or the list of titles currently residing with the OCA terminal. The titles missing from the actual manifest are termed the "delta" or difference between actual and desired manifest.

		After selecting an OCA master using the selection criteria, the OCA terminal will transmit and request the delta list items to the selected OCA(s) in the subnet or cluster in the form of download requests for each title using the URL of the target OCA. This is equivalent to: “terminals adapted to act as relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server.” The function of the two actions are substantially the same, which is to transmit to other selected relay servers or OCAs a modified list of the original transport request or manifest. The way the actions are performed are substantially the same. A modified desired manifest list is a subset list called the “delta.” A relay server (OCA) creates a modified transport request (delta list) of its missing titles and transports this list to another relay server (OCA sends requests to other selected OCAs for the titles on the delta list in a series of requests) and includes addresses of the other relay servers (URLs of the selected OCA terminals in the request as an address to the selected OCAs). The result of these actions are substantially similar: a modification of the original transport request (desired manifest) is transmitted to selected relay terminals (selected OCAs) with addresses (URLs) of the selected relay servers (OCAs).
[9j]	wherein dividing data to be retrieved by said target terminals into a series of packets for transmission to said target terminals and wherein each of said terminals communicates directly with said main server to	Netflix’ OCAs are adapted to communicate with the main server which is hosted by AWS. The OCA’s “Report their status to the Open Connect control panel services in Amazon Web Services”: The Open Connect network can make parallelized cache fill transfers. As such, it must be transferring “packets” of the file. There is evidence that each file is downloaded for cache fill in parallel in packet sizes up to 16 kilobytes. Additionally, the last packet of a download for a cache fill is associated with a notice to the CCS that the download has been completed. The system can also use parallel processes to perform cache transfer using TCP/IP protocols. This is only possible if the CCS is working on a different section of the file at the same time, which is packetized file transfer. There is also evidence that the cache fill transfer uses TCP/IP protocols for packetized data transfer.

	acknowledge receipt of the last packet of a series routed thereto.	TCP data communication protocol, which most of the Internet included Netflix uses, requires an acknowledgement of packets when content is transmitted: TCP is a reliable byte stream delivery service which guarantees that all bytes received will be identical and in the same order as those sent. Since packet transfer by many networks is not reliable, **TCP achieves this using a technique known as positive acknowledgement with re-transmission**. This **requires the receiver to respond with an acknowledgement message as it receives the data**. The sender keeps a record of each packet it sends and maintains a timer from when the packet was sent. The sender re-transmits a packet if the timer expires before receiving the acknowledgement. The timer is needed in case a packet gets lost or corrupted.[Comer, Douglas E. (2006). Internetworking with TCP/IP: Principles, Protocols, and Architecture. Vol. 1 (5th ed.). Prentice Hall. ISBN 978-0-13-187671-2 Netflix requires network traffic to OCA be in TCP protocol:  • **Traffic from OCA**: Allow all destination addresses and ports. • **Traffic to OCA**: Allow TCP 22, 53, 80, 179, 443, UDP 53 and 123 (source and destination), ICMP types 0, 3, 8, 11, and all ICMPv6 from any public IP/port. Allow all return traffic from any appliance-initiated connection (TCP established). https://openconnect.zendesk.com/hc/en-us/articles/360035533071#routi



Glossary of terms

CAGR
Compound Annual Growth Rate

Cache
A temporary local copy of information that originated elsewhere. Thus for CDNs, a copy of files to be delivered to consumers, stored in a local server

CDN (Content Delivery Network)
A distributed system of servers, designed to enable the efficient and reliable distribution of content over the internet

CP (content provider)
An internet business whose focus is delivering content (rather than - say - e-commerce) to consumers. Netflix, CNN and YouTube are examples

FBB
Fixed broadband

Hop
One step in a packet of data's journey across multiple servers

ISP (Internet Service Provider)
A company providing internet connectivity to consumers (consumers or businesses). May provide fixed broadband, mobile data or both

IXP (Internet Exchange Point)
A location where many networks meet to exchange traffic, avoiding the need for multiple bilateral connections

Latency
The lag between a packet of data being sent and reaching its destination

OTT (Over The Top)
Describes services delivered over another network without being integrated with it. YouTube, Facebook and Netflix are examples (since they are not provided by telcos operating broadband networks)

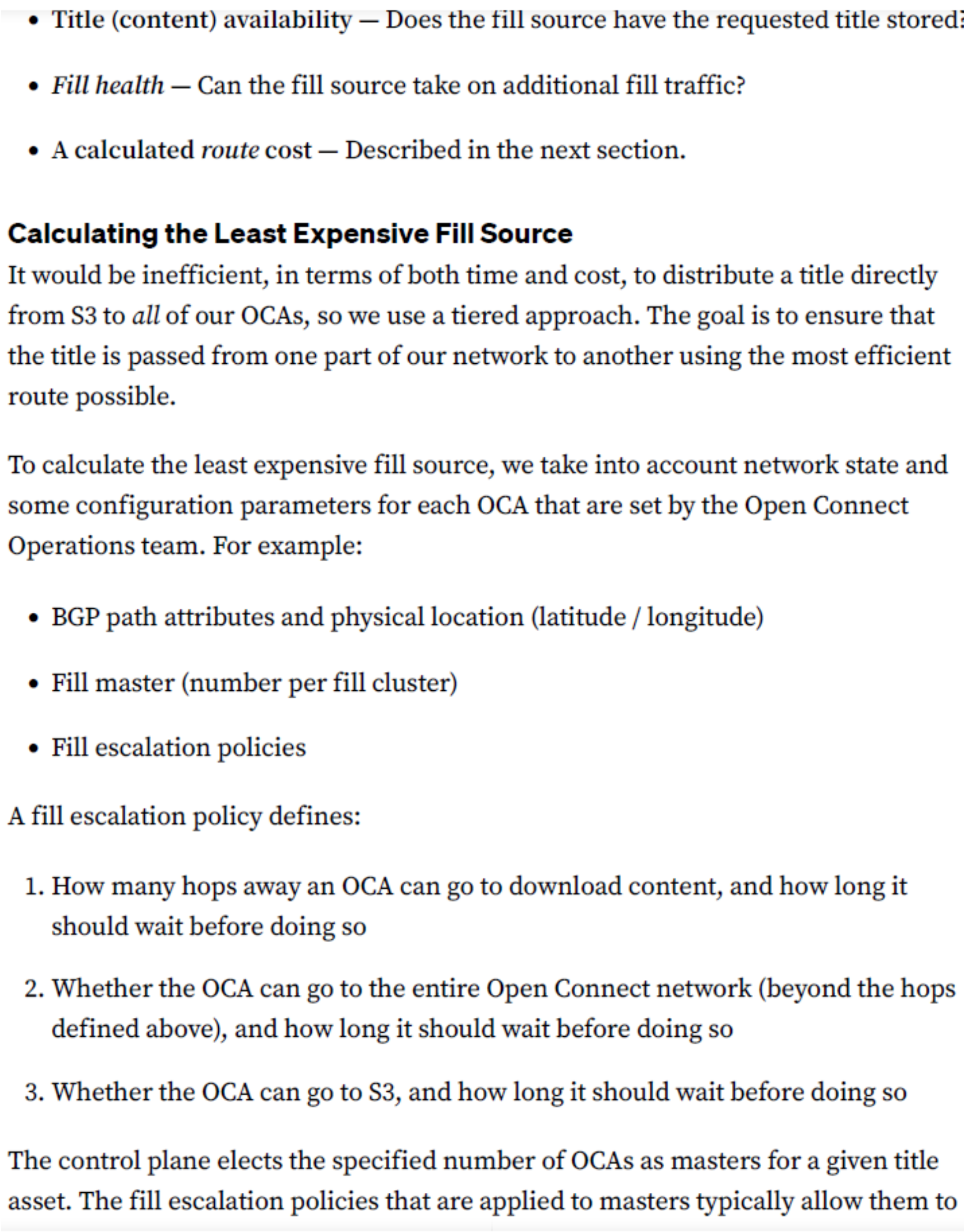
Packet loss
When a router is sent more data than it can handle, it discards a certain amount of data. This is known as packet loss. Typically the data in question will then be requested again from the source server

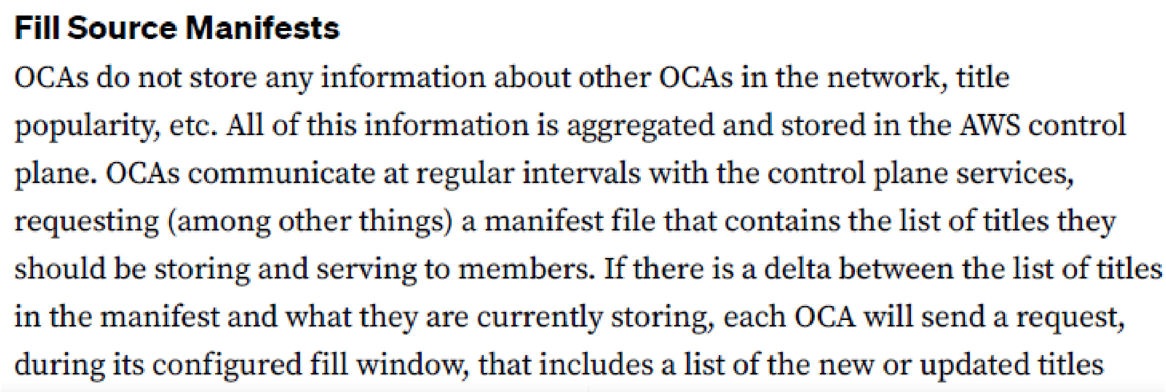
Router
A switch on the internet, that receives packets of data and sends them onwards down the appropriate link

Server
A computer that stores and transmits content

TCP/IP
Transmission control protocol and internet protocol. The two foundational standards for data transmission that underpin the internet

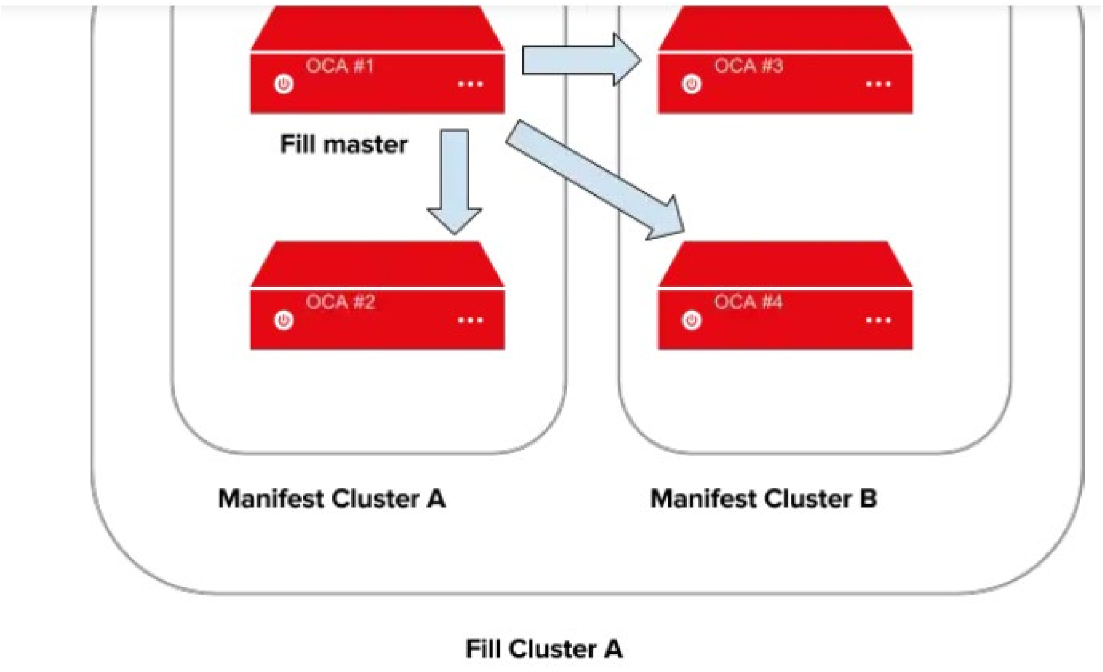
		“A cooperative approach to content delivery,” Netflix (2021), 37. **Where does Netflix use TCP?** Netflix uses TCP for internet streaming to send packets of data for video. Additionally, Netflix specifically looks at the number of TCP connections to determine internet speeds in accordance with testing of OCAs. In another example, “After **the fill master OCA** has completed its S3 download, **it reports back to the control plane** that it now has the title stored.” See https://netflixtechblog.com/netflix-and-fill-c43a32b490c0
[11a]	The method as claimed in claim 9,	*See* above.
[11b]	including operating terminals acting as relay servers to select further downstream target terminals to act as further relay servers on the basis of the relative performances of the further target terminals indicated in said transport request.	OCAs can also download updates from each other – minimizing significant usage of internet “backbone” capacity during the update cycle. The OCAs work in a network to distribute updates among each other and to include further OCAs to which updates and content can be sent. *See* Open Connect Overview, p. 5. The document https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states that there are “**The control plane elects the specified number of OCAs as masters. . . .**” OCAs can act as relay servers, or “masters” that target terminals can use to gain, or fill, content and these are based on relative performance of terminals:

		 • Title (content) availability — Does the fill source have the requested title stored? • *Fill health* — Can the fill source take on additional fill traffic? • A calculated *route* cost — Described in the next section. **Calculating the Least Expensive Fill Source** It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to *all* of our OCAs, so we use a tiered approach. The goal is to ensure that the title is passed from one part of our network to another using the most efficient route possible. To calculate the least expensive fill source, we take into account network state and some configuration parameters for each OCA that are set by the Open Connect Operations team. For example: • BGP path attributes and physical location (latitude / longitude) • Fill master (number per fill cluster) • Fill escalation policies A fill escalation policy defines: 1. How many hops away an OCA can go to download content, and how long it should wait before doing so 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so 3. Whether the OCA can go to S3, and how long it should wait before doing so The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to

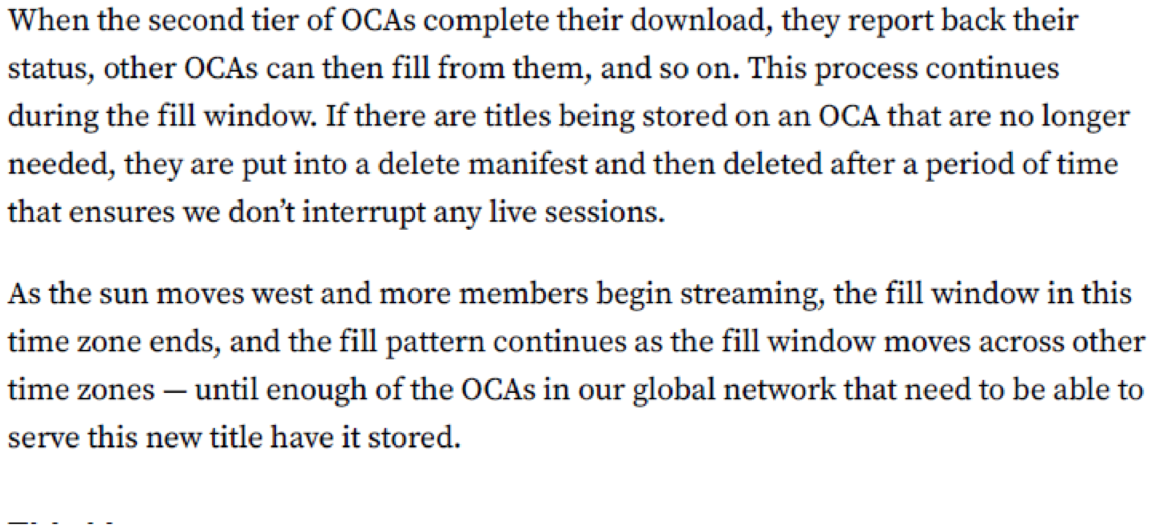
<table>
<tr>
<td></td>
<td></td>
<td>


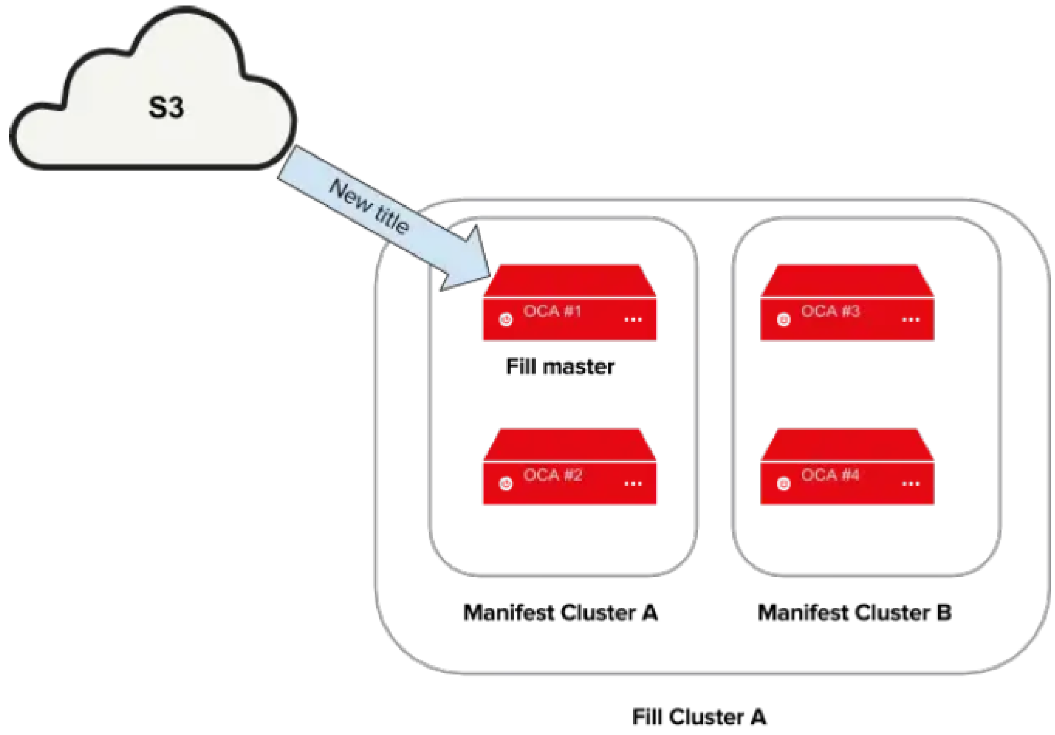
Fill Source Manifests

OCAs do not store any information about other OCAs in the network, title popularity, etc. All of this information is aggregated and stored in the AWS control plane. OCAs communicate at regular intervals with the control plane services, requesting (among other things) a manifest file that contains the list of titles they should be storing and serving to members. If there is a delta between the list of titles in the manifest and what they are currently storing, each OCA will send a request, during its configured fill window, that includes a list of the new or updated titles



</td>
</tr>
</table>

		 When the second tier of OCAs complete their download, they report back their status, other OCAs can then fill from them, and so on. This process continues during the fill window. If there are titles being stored on an OCA that are no longer needed, they are put into a delete manifest and then deleted after a period of time that ensures we don't interrupt any live sessions. As the sun moves west and more members begin streaming, the fill window in this time zone ends, and the fill pattern continues as the fill window moves across other time zones — until enough of the OCAs in our global network that need to be able to serve this new title have it stored.

		1. Peer fill: Available OCAs within the same manifest cluster or the same subnet 2. Tier fill: Available OCAs outside the manifest cluster configuration 3. Cache fill: Direct download from S3 **Example Scenario** After the fill master OCA has completed its S3 download, it reports back to the control plane that it now has the title stored. The next time the other OCAs communicate with the control plane to request a fill source for this title, they are given the option to fill from the fill master.  S3 New title OCA #1 Fill master OCA #2 OCA #3 OCA #4 Manifest Cluster A Manifest Cluster B Fill Cluster A

		Netflix' main server is adapted to send **transport requests** direct to at least one first target terminal on the basis of said terminal performance information. These transport requests are the commands sent by Netflix when its uses "**appliance selection criteria**" ~~and commands where the "**control pane steers clients** to the **best available** OCAs." Netflix' wrote that "**We steer clients to our OCA's based on an ISP's BGP advertisements, coupled with the routing and steering algorithms. . . .**" This steering by Netflix is also included in transport request.~~ Netflix heavily monitors performance and response times for OCAs in a network, as described above. This is so closely monitored by Netflix that it issues a **"Route Performance Report"** to its clients regarding performance of OCAs. The control panel on the main server observes performance data to determine which "second tier" OCAs to complete their downloads with from a master OCA, to "report back their (second tier OCAs) status" to the main server, "and so on" to send transport requests to each new "master" OCA and new "second tier" OCA's, "and so on."
[12a]	The method as claimed in claim 9,	*See* above.
[12b]	wherein the first server is a terminal adapted to act as relay server.	In the Netflix architecture diagram, a terminal can act as the first server, where the other OCAs can have the option of downloading from the "fill master OCA."

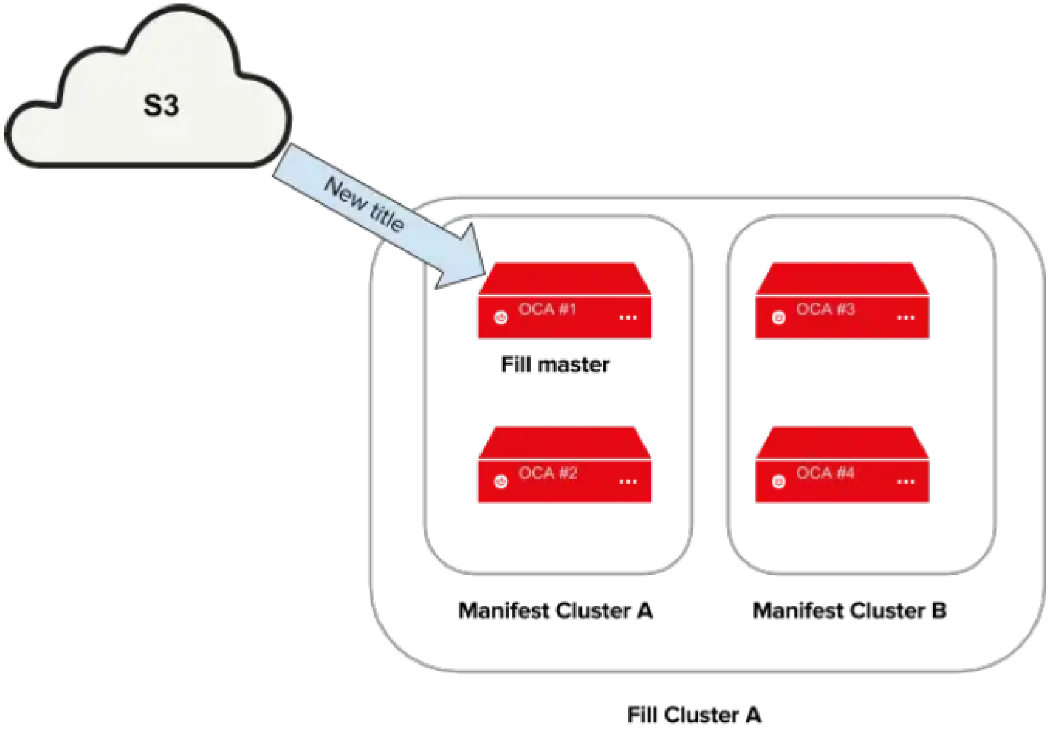
1. Peer fill: Available OCAs within the same manifest cluster or the same subnet

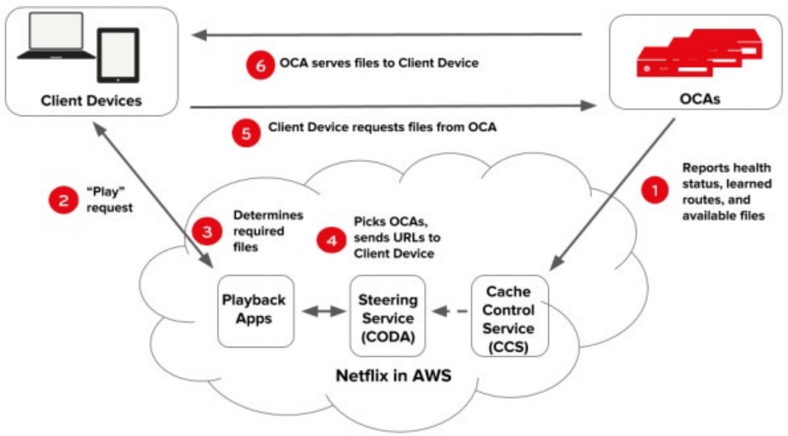
2. Tier fill: Available OCAs outside the manifest cluster configuration

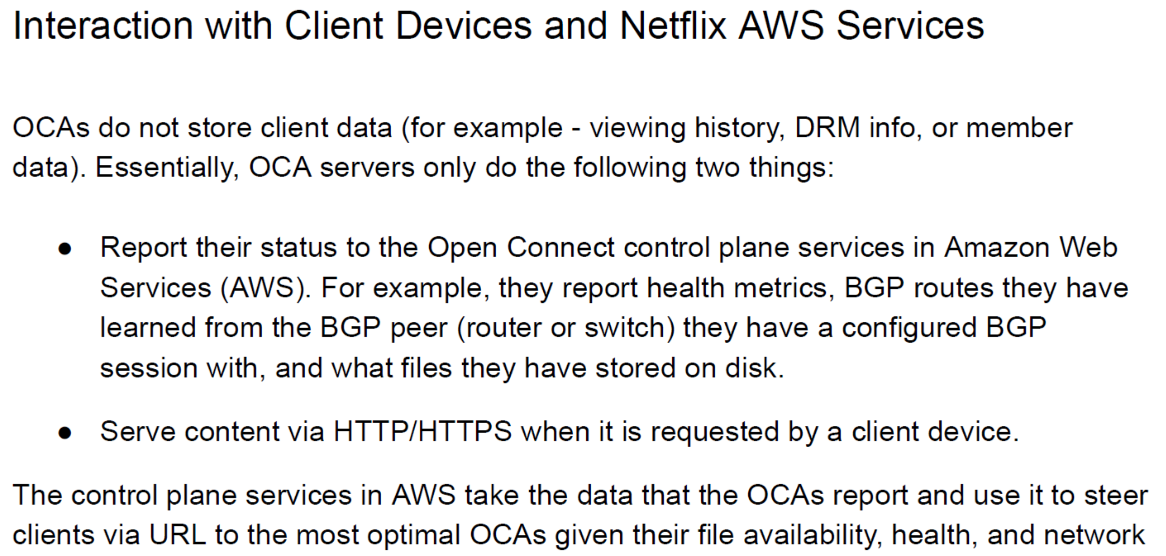
3. Cache fill: Direct download from S3

Example Scenario

After the fill master OCA has completed its S3 download, it reports back to the control plane that it now has the title stored. The next time the other OCAs communicate with the control plane to request a fill source for this title, they are given the option to fill from the fill master.



[13a]	The method as claimed in claim 9,	*See* above.
[13b]	wherein each of said terminals communicates directly with said main server in an upstream direction.	Each of the OCAs can report directly to the Netflix amin server in an upstream direction.  Client Devices OCAs 6 OCA serves files to Client Device 5 Client Device requests files from OCA 1 Reports health status, learned routes, and available files 2 "Play" request 3 Determines required files 4 Picks OCAs, sends URLs to Client Device Playback Apps Steering Service (CODA) Cache Control Service (CCS) Netflix in AWS
[14a]	The method as claimed in claim 9,	*See* above.
[14b]	including routing data to said terminals as routed network protocol traffic such as TCP/IP traffic.	Netflix' OCAs are adapted to communicate with the main server which is hosted by AWS. The OCA's "Report their status to the Open Connect control panel services in Amazon Web Services":

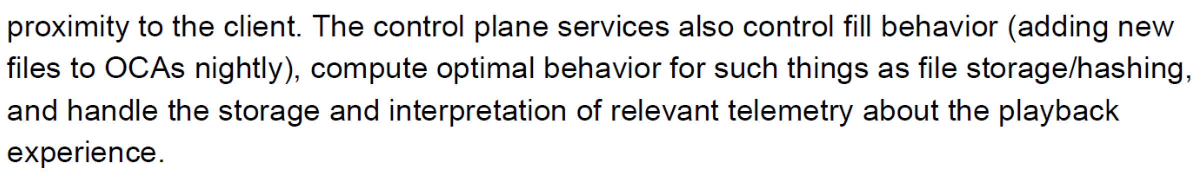


Interaction with Client Devices and Netflix AWS Services

OCAs do not store client data (for example - viewing history, DRM info, or member data). Essentially, OCA servers only do the following two things:

- Report their status to the Open Connect control plane services in Amazon Web Services (AWS). For example, they report health metrics, BGP routes they have learned from the BGP peer (router or switch) they have a configured BGP session with, and what files they have stored on disk.
- Serve content via HTTP/HTTPS when it is requested by a client device.

The control plane services in AWS take the data that the OCAs report and use it to steer clients via URL to the most optimal OCAs given their file availability, health, and network

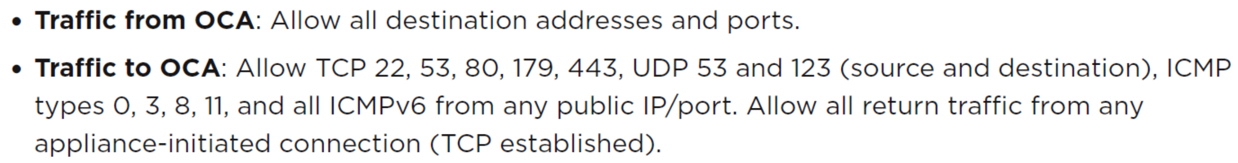


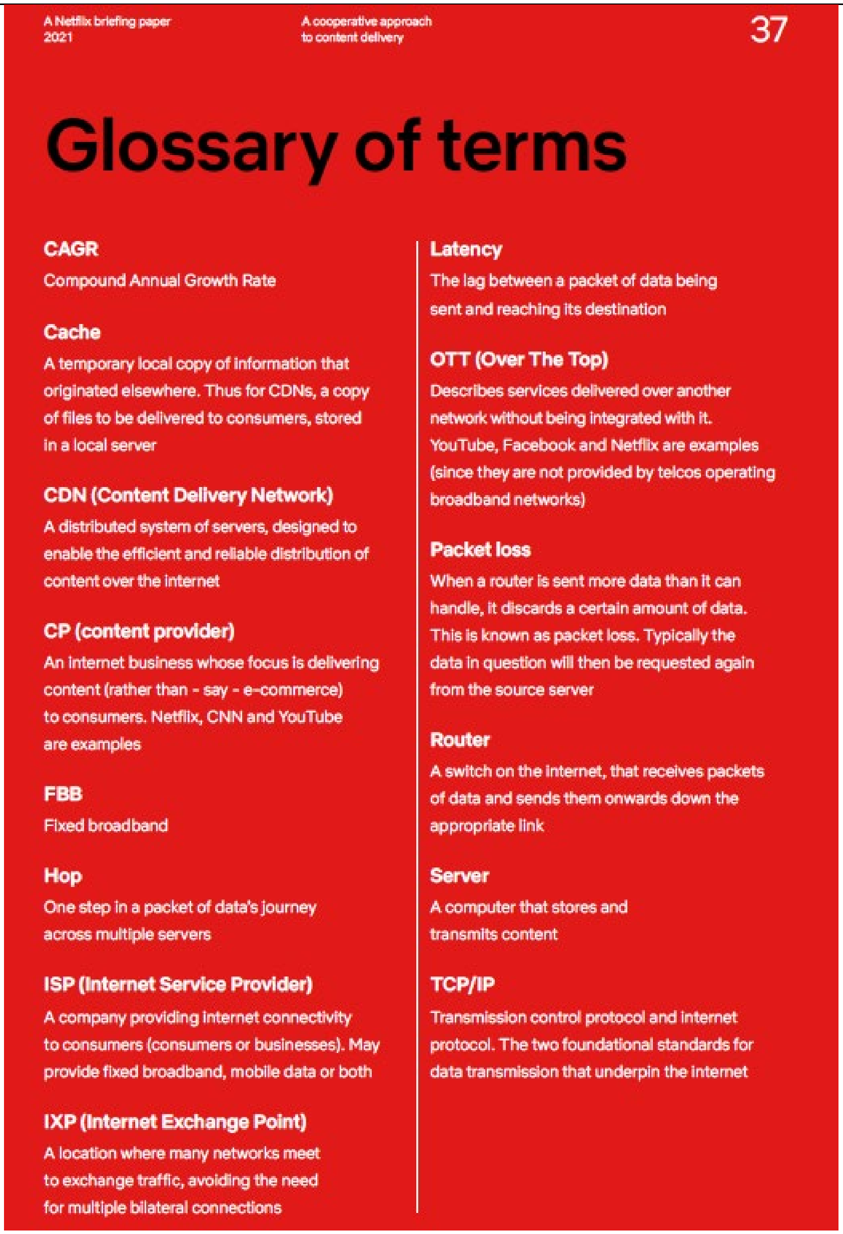
proximity to the client. The control plane services also control fill behavior (adding new files to OCAs nightly), compute optimal behavior for such things as file storage/hashing, and handle the storage and interpretation of relevant telemetry about the playback experience.

See Netflix Open Connect Overview, 4-5.

TCP data communication protocol, which most of the Internet included Netflix uses, requires an acknowledgement of packets when content is transmitted:

TCP is a reliable byte stream delivery service which guarantees that all bytes received will be identical and in the same order as those sent. Since packet transfer by many networks is not reliable, **TCP achieves this using a technique known as positive acknowledgement with re-transmission**. This **requires the receiver to respond with an acknowledgement message as it receives the data**. The sender keeps a record of each packet it sends and maintains a timer from

		when the packet was sent. The sender re-transmits a packet if the timer expires before receiving the acknowledgement. The timer is needed in case a packet gets lost or corrupted.[Comer, Douglas E. (2006). Internetworking with TCP/IP: Principles, Protocols, and Architecture. Vol. 1 (5th ed.). Prentice Hall. ISBN 978-0-13-187671-2 Netflix requires network traffic to OCA be in TCP protocol:  • **Traffic from OCA**: Allow all destination addresses and ports. • **Traffic to OCA**: Allow TCP 22, 53, 80, 179, 443, UDP 53 and 123 (source and destination), ICMP types 0, 3, 8, 11, and all ICMPv6 from any public IP/port. Allow all return traffic from any appliance-initiated connection (TCP established). https://openconnect.zendesk.com/hc/en-us/articles/360035533071#routi



Glossary of terms

CAGR
Compound Annual Growth Rate

Cache
A temporary local copy of information that originated elsewhere. Thus for CDNs, a copy of files to be delivered to consumers, stored in a local server

CDN (Content Delivery Network)
A distributed system of servers, designed to enable the efficient and reliable distribution of content over the internet

CP (content provider)
An internet business whose focus is delivering content (rather than - say - e-commerce) to consumers. Netflix, CNN and YouTube are examples

FBB
Fixed broadband

Hop
One step in a packet of data's journey across multiple servers

ISP (Internet Service Provider)
A company providing internet connectivity to consumers (consumers or businesses). May provide fixed broadband, mobile data or both

IXP (Internet Exchange Point)
A location where many networks meet to exchange traffic, avoiding the need for multiple bilateral connections

Latency
The lag between a packet of data being sent and reaching its destination

OTT (Over The Top)
Describes services delivered over another network without being integrated with it. YouTube, Facebook and Netflix are examples (since they are not provided by telcos operating broadband networks)

Packet loss
When a router is sent more data than it can handle, it discards a certain amount of data. This is known as packet loss. Typically the data in question will then be requested again from the source server

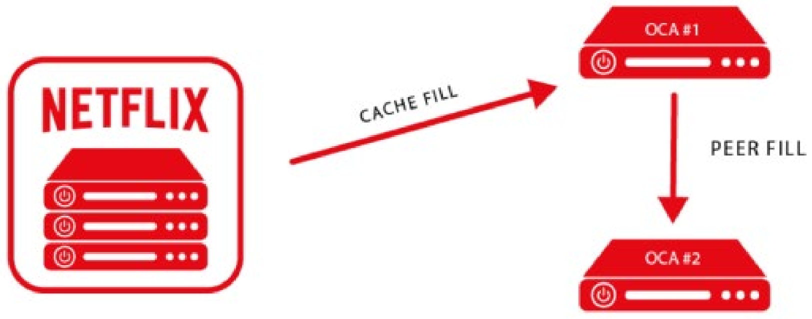
Router
A switch on the internet, that receives packets of data and sends them onwards down the appropriate link

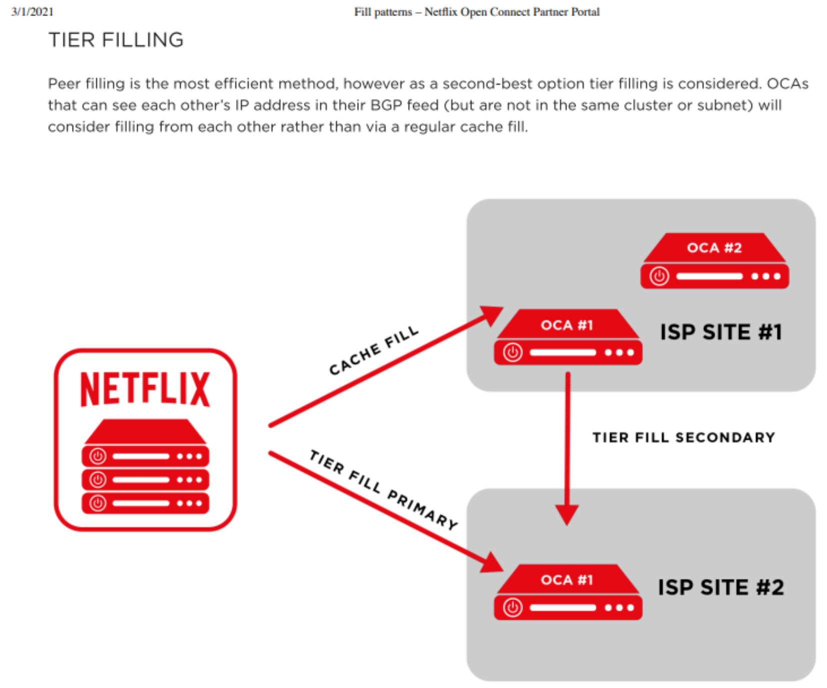
Server
A computer that stores and transmits content

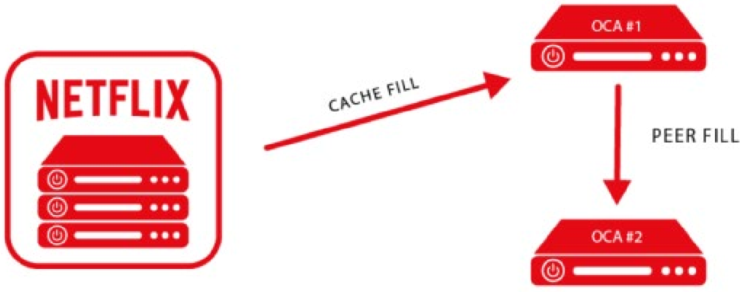
TCP/IP
Transmission control protocol and internet protocol. The two foundational standards for data transmission that underpin the internet

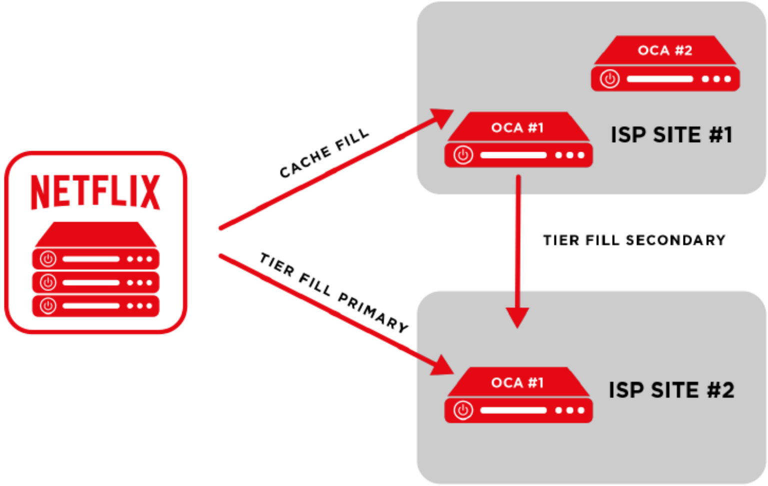
		“A cooperative approach to content delivery,” Netflix (2021), 37. Where does Netflix use TCP? Netflix uses TCP for internet streaming to send packets of data for video. Additionally, Netflix specifically looks at the number of TCP connections to determine internet speeds in accordance with testing of OCAs. In another example, “After **the fill master OCA** has completed its S3 download, **it reports back to the control plane** that it now has the title stored.” See https://netflixtechblog.com/netflix-and-fill-c43a32b490c0
[15a]	The method as claimed in claim 9,	*See* above.
[15b]	including transmitting said data in binary format.	Netflix states, on its public technical blog, that binary format data can be used in image file formats sent through the network. *See* https://netflixtechblog.com/enhancing-the-netflix-ui-experience-with-hdr-1e7506ad3e8
[16a]	A network server adapted to operate as a main server in a data communication network, the data communication network including:	Netflix uses a system called **Open Connect** to deliver Netflix TV shows and movies to members world-wide. Netflix runs the operation of Open Connect from a Netflix application (CCS server) that is hosted in AWS. *See* Open Connect Overview, p. 4-5. The building blocks of Open Connect are our suite of purpose-built **server appliances, called Open Connect Appliances (OCAs)**. *See* Open Connect Overview, p. 2. These are deployed directly inside ISP networks. Netflix provides the server hardware. Netflix runs the cache fill operation of Open Connect from a Netflix CCS (main server) that is hosted in AWS. *See* Open Connect Overview, p. 4-5.

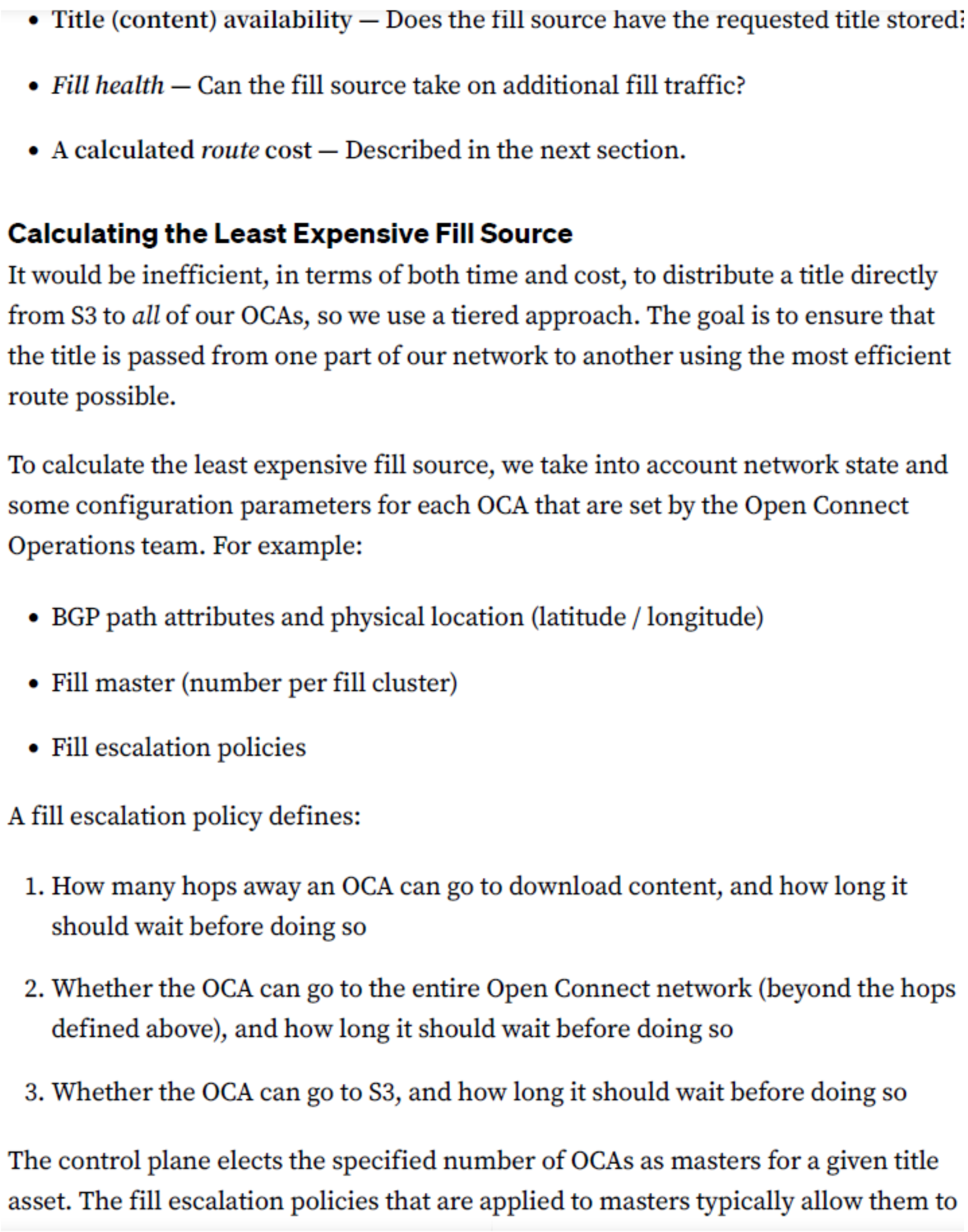
[16b]	a plurality of terminals, a network information database and a first server which from which data be retrieved by at least one target terminal from among said plurality of terminals, at least two of said terminals being adapted to act as relay servers for serving data retrieved from said first server to at least one further target terminal based on terminal performance information stored in the network information database, said network server being distinct from said first server;	Netflix runs the operation of Open Connect from a Netflix application (CCS server) that is hosted in AWS. *See* Open Connect Overview, p. 4-5. In its global network, Netflix provides **data centers** such as an "S3" server **("first server")** housing **content ("data") on at lease one server**, and provides OCA users (such as ISP's) direct access to these data centers over the Internet that are housing the content. One or more of these **data centers** house a "**first server**" according to the claims. In deployment of Open Connect, Netflix provides Internet Service Providers with an OCA appliance direct "settlement-free interconnection (SFI)." The terminal OCA can "Connect via direct Private Network Interconnect (PNI) or IXP-based SFI peering to Netflix Open Connect Appliances **in our data centers**." "Netflix has the ability to interconnect at a number of global data center facilities and public Internet Exchange fabrics as listed on our Peering Locations page. We openly peer with any network at IXP locations where we are mutually present and we consider private interconnection as appropriate." ISPs who do not currently participate in pubic peering might want to consider that a single IX port can support multiple peering sessions, providing **direct access to various content,** cloud, and network providers. Welcome to Open Connect, p. 3. Dkt 39 at p. 48. The following diagram also illustrates access from a target terminal (OCA #1, OCA #2) to a Netflix first server in "our data centers." OCAs in a cluster and on the same subnet can attempt peer filling from each other. There is also Tier filling where if in different ISP sites. deploymentguide.pdf (netflix.com)

		 https://openconnect.zendesk.com/hc/en-us/articles/360035618071 2/4

		 3/1/2021 Fill patterns – Netflix Open Connect Partner Portal TIER FILLING Peer filling is the most efficient method, however as a second-best option tier filling is considered. OCAs that can see each other's IP address in their BGP feed (but are not in the same cluster or subnet) will consider filling from each other rather than via a regular cache fill. NETFLIX CACHE FILL TIER FILL PRIMARY TIER FILL SECONDARY OCA #2 OCA #1 ISP SITE #1 OCA #1 ISP SITE #2 The "target terminal selected form said plurality of terminals" language of the claim is infringed by the health and performance monitoring and OCA target terminal selection process described below in Sec. [1e].
[16c]	said network server being adapted to manage selective retrieval of data from said first server by at least one target terminal selected from said plurality of	A "first server" is identified above in Sec. [1a] as a server within Netflix' data centers. According to Netflix' network architecture, OCAs in a cluster and on the same subnet can attempt peer filling content "cache fill" from the first server (data center) to OCA #1. OCA#1 then acts as a "relay server" to fill cached content to each other, from OCA#1 to OCA#2. OCA#2 becomes a "target terminal" in this example that is served data from the relay server (OCA#1) retrieved from the first server (data center). See https://openconnect.netflix.com/deploymentguide.pdf.

	terminals; and wherein said network server being further adapted to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis of their relative response times,	 https://openconnect.zendesk.com/hc/en-us/articles/360035618071 2/4

		In another example, Netflix's first server (data center) can cache fill content to a terminal OCA#1, which then acts as a "relay server" to fill a target terminal (OCA#2) using "tier filling." "Peer filling is the most efficient method, however as a second-best option tier filling is considered. OCAs that can see each other's IP address in their BGP feed (but are not in the same cluster or subnet) will consider filling from each other rather than via a regular cache fill."  NETFLIX CACHE FILL TIER FILL PRIMARY TIER FILL SECONDARY OCA #2 OCA #1 ISP SITE #1 OCA #1 ISP SITE #2 See https://openconnect.netflix.com/deploymentguide.pdf. In another example, https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states that there are "**The control plane elects the specified number of OCAs as masters. . . .**" OCAs can act as relay servers, or "masters" that target terminals can use to gain, or fill, content:



- Title (content) availability — Does the fill source have the requested title stored?
- *Fill health* — Can the fill source take on additional fill traffic?
- A calculated *route* cost — Described in the next section.

Calculating the Least Expensive Fill Source

It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to *all* of our OCAs, so we use a tiered approach. The goal is to ensure that the title is passed from one part of our network to another using the most efficient route possible.

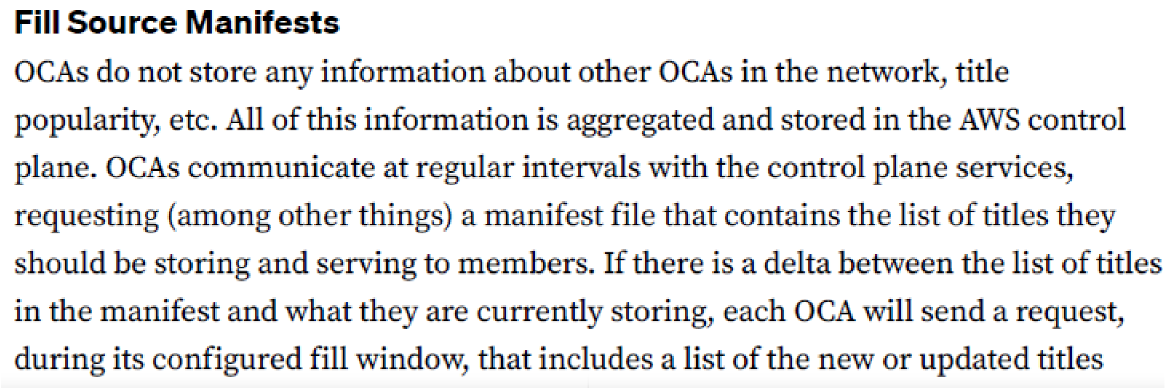
To calculate the least expensive fill source, we take into account network state and some configuration parameters for each OCA that are set by the Open Connect Operations team. For example:

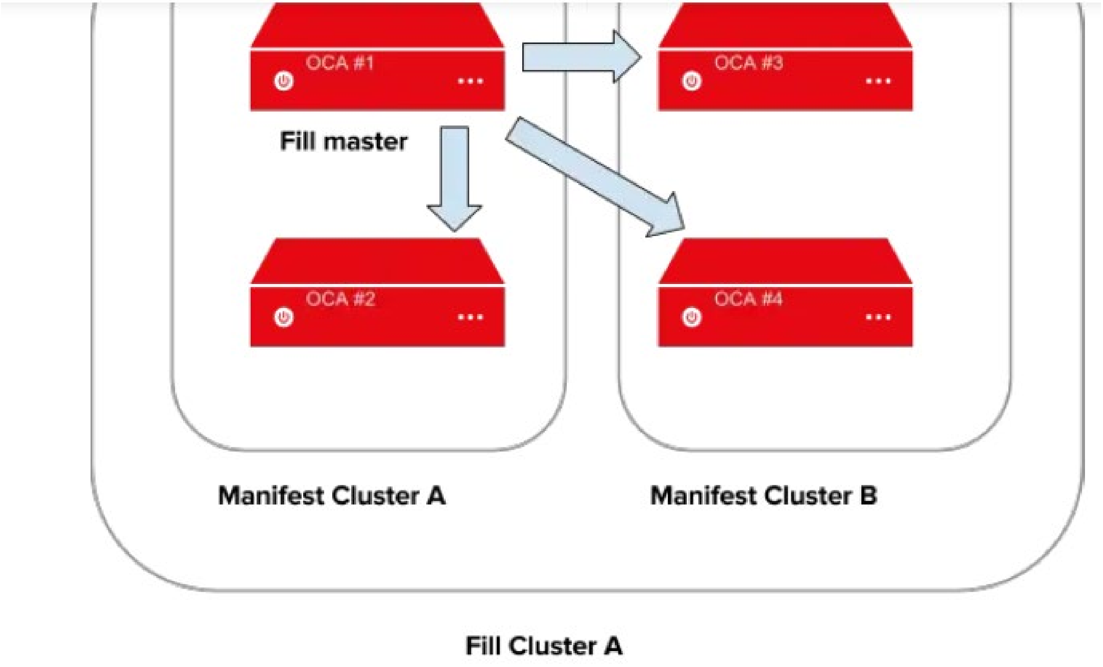
- BGP path attributes and physical location (latitude / longitude)
- Fill master (number per fill cluster)
- Fill escalation policies

A fill escalation policy defines:

1. How many hops away an OCA can go to download content, and how long it should wait before doing so
2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so
3. Whether the OCA can go to S3, and how long it should wait before doing so

The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to

		The CCS server monitors the “fill health” of each potential download locations, or fill sources that includes OCAs, and uses a fill escalation policy based upon response times to determine if the performance of the download location can take on additional traffic or not and how long the OCA should wait for a response before escalation to the next download locations.  **Fill Source Manifests** OCAs do not store any information about other OCAs in the network, title popularity, etc. All of this information is aggregated and stored in the AWS control plane. OCAs communicate at regular intervals with the control plane services, requesting (among other things) a manifest file that contains the list of titles they should be storing and serving to members. If there is a delta between the list of titles in the manifest and what they are currently storing, each OCA will send a request, during its configured fill window, that includes a list of the new or updated titles

		OCA #1 OCA #3 Fill master OCA #2 OCA #4 Manifest Cluster A Manifest Cluster B Fill Cluster A

		When the second tier of OCAs complete their download, they report back their status, other OCAs can then fill from them, and so on. This process continues during the fill window. If there are titles being stored on an OCA that are no longer needed, they are put into a delete manifest and then deleted after a period of time that ensures we don’t interrupt any live sessions. As the sun moves west and more members begin streaming, the fill window in this time zone ends, and the fill pattern continues as the fill window moves across other time zones — until enough of the OCAs in our global network that need to be able to serve this new title have it stored.

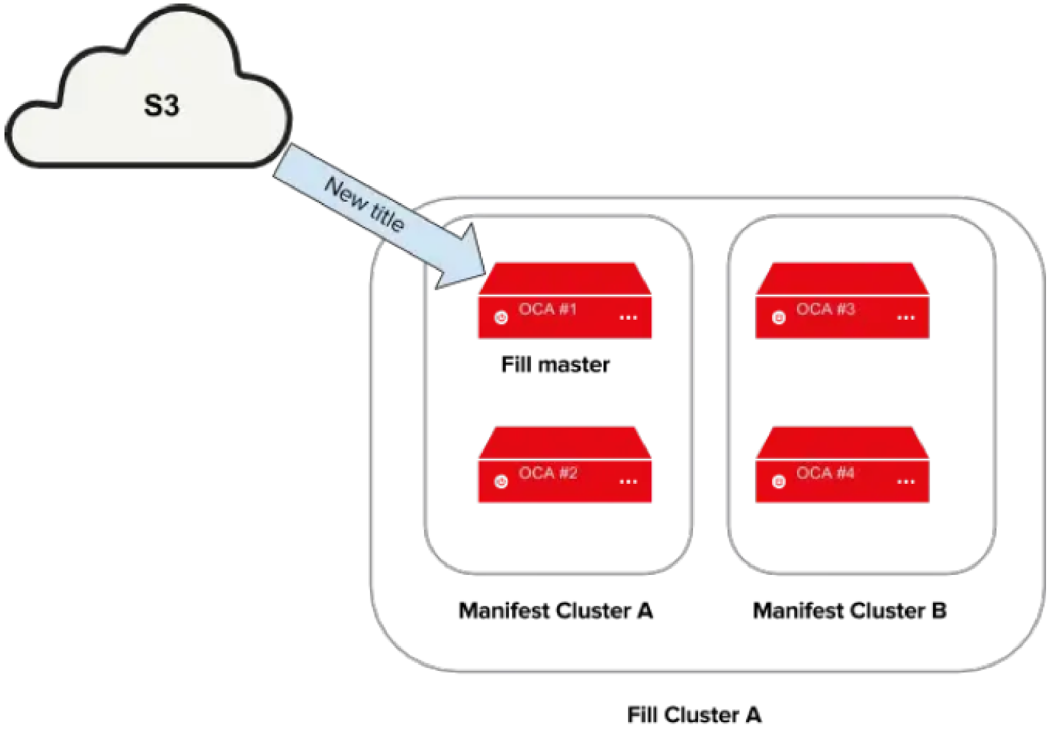
1. Peer fill: Available OCAs within the same manifest cluster or the same subnet

2. Tier fill: Available OCAs outside the manifest cluster configuration

3. Cache fill: Direct download from S3

Example Scenario

After the fill master OCA has completed its S3 download, it reports back to the control plane that it now has the title stored. The next time the other OCAs communicate with the control plane to request a fill source for this title, they are given the option to fill from the fill master.



[16d]	said network server being further adapted to send transport requests direct to at least one first target terminal that is adapted to act as a relay server, each such transport request includes details of data to be retrieved, the address of the first server from which the data is to be requested by the first target terminal, the addresses of at least one second target terminal to which the data retrieved from the first server is to be relayed by the first target terminal and an indication of a relative performance of a further target terminal based on	All OCA deployments are constantly monitored to ensure reliability and efficiency. Netflix **makes use of non-peak bandwidth to download the vast majority of content updates to the OCAs in network during these configurable time windows. OCAs can also download updates from each other – minimizing significant usage of internet “backbone” capacity during the update cycle.** A “desired manifest” and when needed an “emergency manifest” are posted by Netflix to the CCS server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. A “desired manifest” and when needed an “emergency manifest” are posted by Netflix to the CCS (Control Content Service) server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. Equivalently, these manifests are fetched by regularly from the CCS by each OCA terminal, which traverses the desired manifest or emergency manifest. The CCS server acts as an email inbox for manifests sent directly to the OCAs, where each OCA terminal is mandated to check regularly for manifests and download the manifests to the local OCA terminal hard drive. This desired manifest and emergency manifest and the download location data on the CCS are equivalent the “server is adapted to send transport requests direct to at least one first target terminal.” The functions are substantially the same - to send a request from a main server (CCS) to a terminal (OCA) with instructions to download or “fill” content. The way this is accomplished is substantially the same. The claim states that the main server is adapted to send transport requests direct to at least one first target terminal over a network, and on the Open Connect Network, Netflix posts a desired manifest or emergency manifest to the CCS server that is intended for each OCA in the network. The results are substantially the same - to transmit commands from the main server/CCS server to a terminal/OCA to download content data in the manifest that needs to be retrieved.

	the terminal performance information stored in the network information database;	The control plane elects the specified number of OCAs as masters for a given title asset. https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 Thus, the CCS server selects OCA terminals as a download location for a given title asset. The main server on CCS is further adapted to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis of their relative response times. At https://netflixtechblog.com/netflix-and-fill-c43a32b490c0, Netflix wrote: The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title. The determination of the list takes into consideration several high-level factors: Title (content) availability — Does the fill source have the requested title stored? **Fill health — Can the fill source take on additional fill traffic?** A calculated route cost — Described in the next section. A fill escalation policy defines: 10. How many hops away an OCA can go to download content, and **how long it should wait before doing so** 11. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), **and how long it should wait before doing so** 12. Whether the OCA can go to S3, **and how long it should wait before doing so** (Emphasis added.) The CCS server monitors the "fill health" of each potential download locations, or fill sources that includes OCAs, and uses a fill escalation policy based upon response times to determine if the performance of the download location can take on additional traffic or not and how long the OCA should wait for a response before escalation to the next download locations. Thus, the manifest is assembled by the CCS server and download locations are assembled based on OCA performance

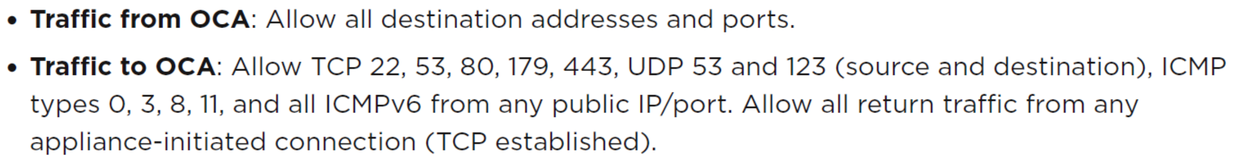
		information, and the manifest is sent to a given OCA. Therefore, the transport request is sent on a basis of said terminal performance information. A “desired manifest” and when needed an “emergency manifest” are transport requests posted by Netflix to the CCS (Control Content Service) server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. these manifests are fetched by regularly from the CCS by each OCA terminal, which traverses the desired manifest or emergency manifest. OCAs then then query the CCS for location information files that list where each title on the desired manifest that is needed by an OCA can be downloaded. https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states a master OCA and at least one second OCA can be selected based on their relative performance: The CCS server information location list provides the address of a first server, called the “S3” server, for a download location and provides other download locations of a second and additional master OCAs: It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to all of our OCAs, so we use a tiered approach. . . . A fill escalation policy defines: 1. How many hops away an OCA can go to download content, and how long it should wait before doing so 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so 3. Whether the OCA can go to S3, and how long it should wait before doing so The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to reach farther with less delay in order to grab that content and then share it locally with non-masters.

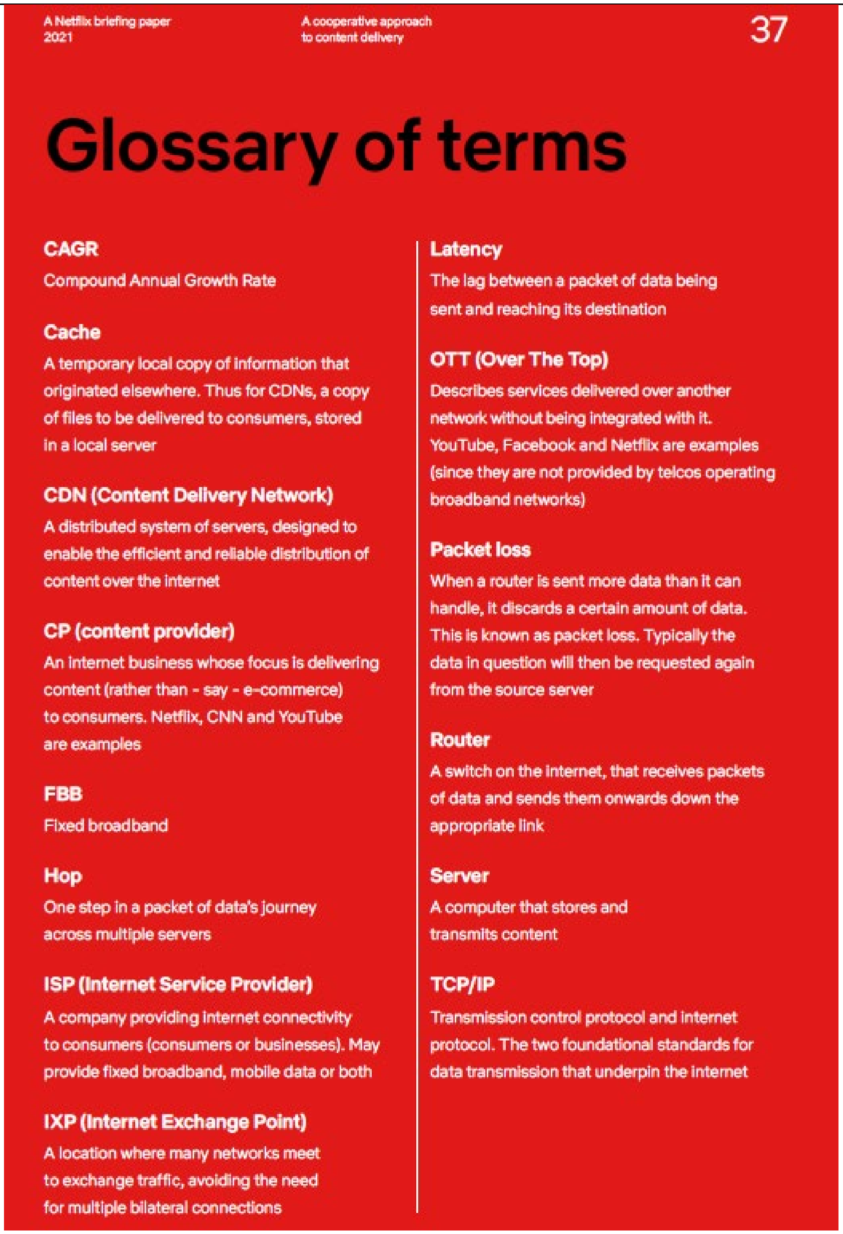
		As stated in [1e], the CCS monitors the "fill health" and performance of download locations, which is based on performance of the OCAs, to determine if that OCA will be selected as a download location or not. This desired manifest and emergency manifest along with the downloaded location information files and fill policy for master OCAs and a second (target terminal) OCA to fill from a (first terminal) OCA on the CCS are equivalent the "each such transport request includes details of data to be retrieved, the address of the first server from which the data is to be requested by the first target terminal, the addresses of at least one second target terminal to which the data from the first server to be relayed by the first target terminal." The functions are substantially the same - to send a request from a main server (CCS) to a terminal (OCA) with instructions to download or "fill" content from specific master terminal/OCA addresses and includes an address of at least one second terminal/OCA. The way this is accomplished is substantially the same. The claim states that the main server is adapted to send transport requests that include download locations direct to at least one first target terminal over a network, and on the Open Connect Network, Netflix posts a desired manifest or emergency manifest to the CCS server along with download locations that are intended for each OCA to read on a regular basis. The results are substantially the same - to transmit commands from the main server/CCS server to a terminal/OCA to download content data in the manifest that needs to be retrieved from various addresses.
[16e]	wherein terminals adapted to act as relay servers are adapted to modify transport requests received from said network server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target	OCAs can also download updates from each other – minimizing significant usage of internet "backbone" capacity during the update cycle. The OCAs work in a network to distribute updates among each other and to include further OCAs to which updates and content can be sent. *See* Open Connect Overview, p. 5; Fill Patterns, pp. 1-3. Netflix' OCA that are adapted to act as relay servers (see 1d above) are adapted to modify transport requests received from the main server or from other relay servers and transmit the modified transport request to selected target terminals that includes addresses of further target terminals. The CCS server will to order OCA terminals to peer or tier fill from using OCAs selected by the CCS server. The CCS server uses Appliance Section Criteria to select OCA terminals as targets in the to transfer titles from: 1) the terminal appliance that receives the most-specific route; 2) the terminal appliance that receives the route to the client's netblock with the shortest AS path; 3) the terminal appliance that receives the route to the client's netblock with the lowest multi-exit

	terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server; and	discriminator; 4) the geographically closest appliance. The CCS server includes the URL addresses of these master or target terminals in the desired manifest, which is loaded by an OCA terminal in its memory or hard drive space in order to select an OCA for downloading titles from. After receiving the desired manifest, an OCA terminal will compare the manifest with its own "actual manifest", or the list of titles currently residing with the OCA terminal. The titles missing from the actual manifest are termed the "delta" or difference between actual and desired manifest. The OCA terminal will then query the CCS terminal for a list of download locations for each title on the delta. The CCS responds, as stated in [1f] with a list of URLs that are downloadable locations of master OCAs for each individual title needed by an OCA to fill its delta: "OCAs communicate at regular intervals with the control plane services, requesting (among other things) a manifest file that contains the list of titles they should be storing and serving to members. If there is a **delta** between the list of titles in the manifest and what they are currently storing, each OCA will send a request, during its configured fill window, **that includes a list of the new or updated titles that it needs**. The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title." (Emphasis added.) *See* https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 The action of an OCA requesting download locations (master OCAs) for its delta list from the CCS, and then requesting a delta-listed title from the list of a master OCAs, is equivalent to a modified transport request. The OCA is using a modified list of titles (delta or missing titles list from the desired manifest) to request a title or titles from further target terminals, or master OCAs. This is equivalent to: "terminals adapted to act as relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server." The function of the two actions are substantially the same, which is to transmit to other selected relay servers or OCAs a modification of the original transport request, or desired manifest, along with URLs of those relay servers or URLs of master OCAs. The way the actions are performed are substantially the same. A delta is a modification of the manifest list, or in other words a subset list

		of what it is supposed to download. Instead of the terminal, or OCA, transmitting the delta list to another OCA, an "actual manifest" is sent to the CCS server which responds to the OCA with a list of URL locations to download the modified list titles of its delta list, after which the delta list is then sent to another OCA via a series of modified transfer requests. The result of these actions are substantially similar: further relay terminal addresses are sent to the OCAs, a modification of the original transport request (desired manifest) is transmitted to selected relay terminals (single or multiple delta titles to selected OCAs) are sent to addresses (URLs) of further selected relay servers (OCAs) in the form of a request for one or more titles from one or more master OCAs. Alternatively and equivalently, Netflix documentation discloses that OCA terminals, if they are clustered or if they are in the same subnet, will attempt to peer or tier fill from each other. https://openconnect.zendesk.com/hc/en-us/articles/360035618071-Fill-patterns OCA terminals in a subnet or cluster broadcast their IP and physical locations to one another and save this information. In general, appliances determine where to receive fill using selection criteria that is used by Netflix client devices. The OCA terminals then use a similar Appliance Section Criteria as the CCS server uses to select OCA terminals as targets in the subnet or cluster to transfer titles from: 1) the terminal appliance that receives the most-specific route; 2) the appliance terminal that receives the route to the client's netblock with the shortest AS path; 3) the appliance terminal that receives the route to the client's netblock with the lowest multi-exit discriminator; 4) the geographically closest terminal. The OCA terminal includes the URL addresses of these terminals in its memory or hard drive space in order to select an OCA for downloading titles from. See *Fill Patterns*, pp. 1-4. After receiving the desired manifest, an OCA terminal will compare the manifest with its own "actual manifest", or the list of titles currently residing with the OCA terminal. The titles missing from the actual manifest are termed the "delta" or difference between actual and desired manifest. After selecting an OCA master using the selection criteria, the OCA terminal will transmit and request the delta list items to the selected OCA(s) in the subnet or cluster in the form of download requests for each title using the URL of the target OCA. This is equivalent to: "terminals adapted to act as relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport

		request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server." The function of the two actions are substantially the same, which is to transmit to other selected relay servers or OCAs a modified list of the original transport request or manifest. The way the actions are performed are substantially the same. A modified desired manifest list is a subset list called the "delta." A relay server (OCA) creates a modified transport request (delta list) of its missing titles and transports this list to another relay server (OCA sends requests to other selected OCAs for the titles on the delta list in a series of requests) and includes addresses of the other relay servers (URLs of the selected OCA terminals in the request as an address to the selected OCAs). The result of these actions are substantially similar: a modification of the original transport request (desired manifest) is transmitted to selected relay terminals (selected OCAs) with addresses (URLs) of the selected relay servers (OCAs).
[16f]	wherein data to be retrieved by said target terminals are divided into a series of packets for transmission to said target terminals and each of said terminals are adapted to communicate directly with said main server to acknowledge receipt of the last packet of a series routed thereto.	Netflix' OCAs are adapted to communicate with the main server which is hosted by AWS. The OCA's "Report their status to the Open Connect control panel services in Amazon Web Services": The Open Connect network can make parallelized cache fill transfers. As such, it must be transferring "packets" of the file. There is evidence that each file is downloaded for cache fill in parallel in packet sizes up to 16 kilobytes. Additionally, the last packet of a download for a cache fill is associated with a notice to the CCS that the download has been completed. The system can also use parallel processes to perform cache transfer using TCP/IP protocols. This is only possible if the CCS is working on a different section of the file at the same time, which is packetized file transfer. There is also evidence that the cache fill transfer uses TCP/IP protocols for packetized data transfer. TCP data communication protocol, which most of the Internet included Netflix uses, requires an acknowledgement of packets when content is transmitted: TCP is a reliable byte stream delivery service which guarantees that all bytes received will be identical and in the same order as those sent. Since packet transfer by many networks is not reliable,

		TCP achieves this using a technique known as positive acknowledgement with re-transmission. This **requires the receiver to respond with an acknowledgement message as it receives the data**. The sender keeps a record of each packet it sends and maintains a timer from when the packet was sent. The sender re-transmits a packet if the timer expires before receiving the acknowledgement. The timer is needed in case a packet gets lost or corrupted.[Comer, Douglas E. (2006). Internetworking with TCP/IP: Principles, Protocols, and Architecture. Vol. 1 (5th ed.). Prentice Hall. ISBN 978-0-13-187671-2 Netflix requires network traffic to OCA be in TCP protocol:  • **Traffic from OCA**: Allow all destination addresses and ports. • **Traffic to OCA**: Allow TCP 22, 53, 80, 179, 443, UDP 53 and 123 (source and destination), ICMP types 0, 3, 8, 11, and all ICMPv6 from any public IP/port. Allow all return traffic from any appliance-initiated connection (TCP established). https://openconnect.zendesk.com/hc/en-us/articles/360035533071#routi



Glossary of terms

CAGR
Compound Annual Growth Rate

Cache
A temporary local copy of information that originated elsewhere. Thus for CDNs, a copy of files to be delivered to consumers, stored in a local server

CDN (Content Delivery Network)
A distributed system of servers, designed to enable the efficient and reliable distribution of content over the internet

CP (content provider)
An internet business whose focus is delivering content (rather than - say - e–commerce) to consumers. Netflix, CNN and YouTube are examples

FBB
Fixed broadband

Hop
One step in a packet of data's journey across multiple servers

ISP (Internet Service Provider)
A company providing internet connectivity to consumers (consumers or businesses). May provide fixed broadband, mobile data or both

IXP (Internet Exchange Point)
A location where many networks meet to exchange traffic, avoiding the need for multiple bilateral connections

Latency
The lag between a packet of data being sent and reaching its destination

OTT (Over The Top)
Describes services delivered over another network without being integrated with it. YouTube, Facebook and Netflix are examples (since they are not provided by telcos operating broadband networks)

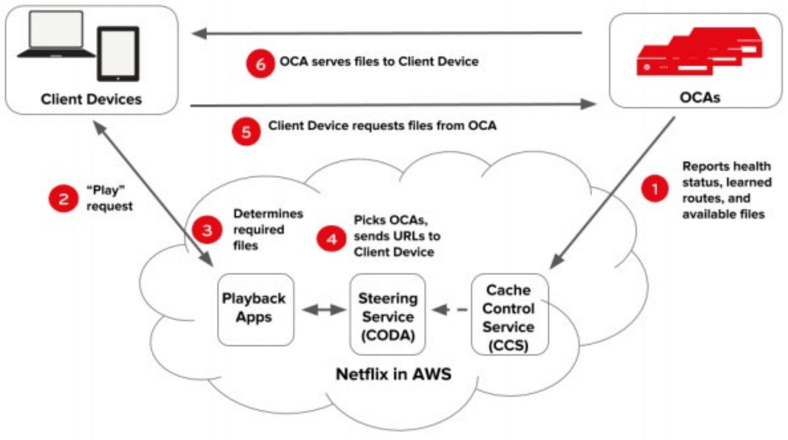
Packet loss
When a router is sent more data than it can handle, it discards a certain amount of data. This is known as packet loss. Typically the data in question will then be requested again from the source server

Router
A switch on the internet, that receives packets of data and sends them onwards down the appropriate link

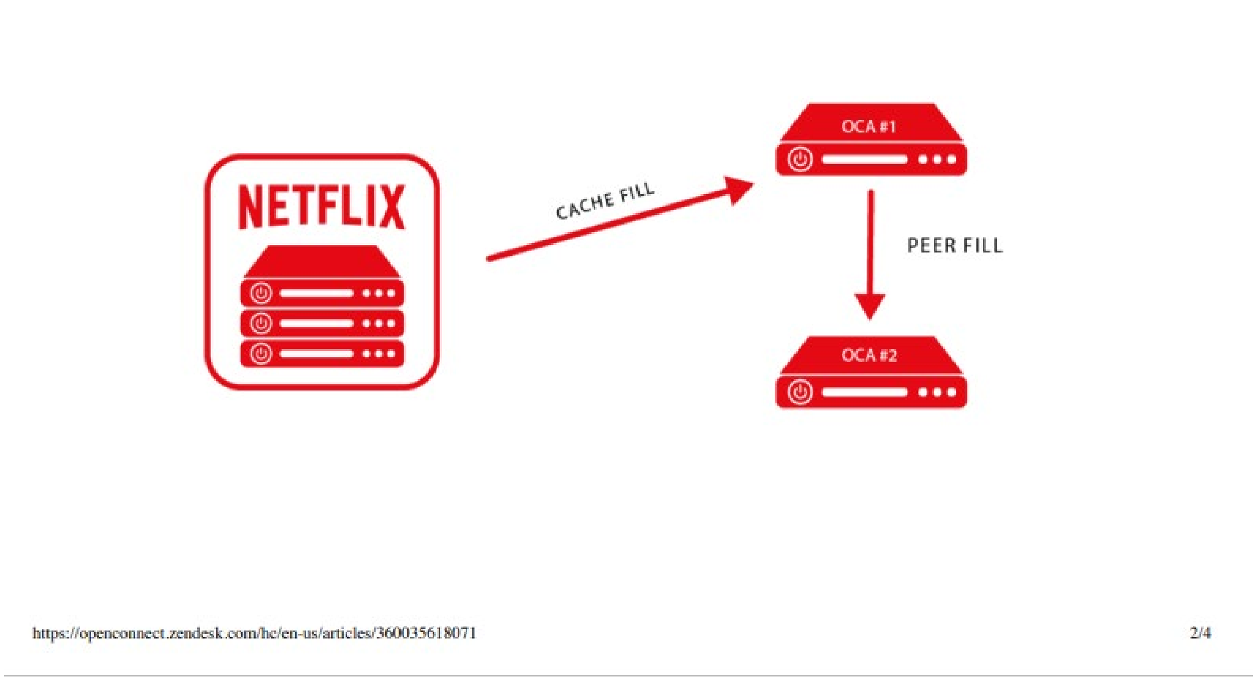
Server
A computer that stores and transmits content

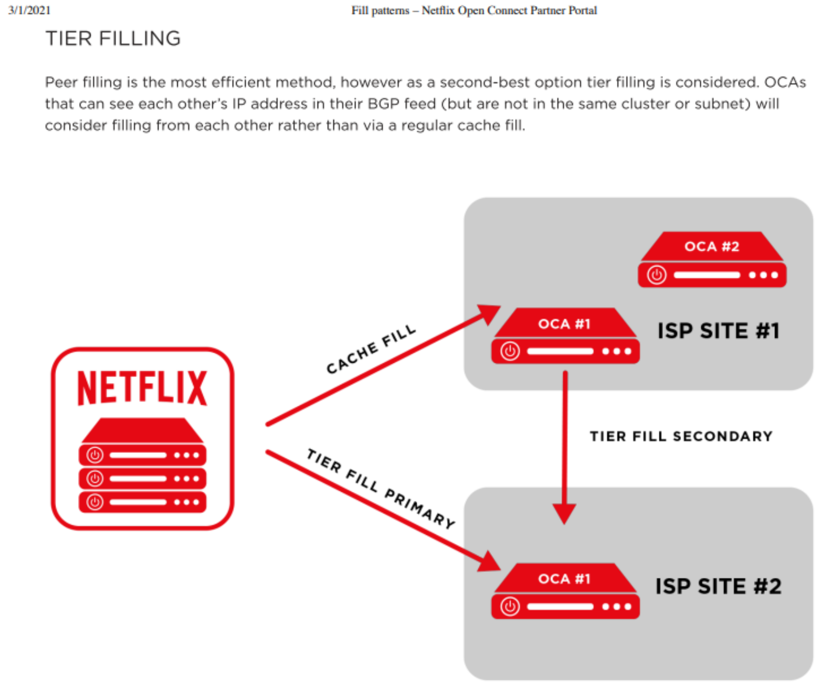
TCP/IP
Transmission control protocol and internet protocol. The two foundational standards for data transmission that underpin the internet

		"A cooperative approach to content delivery," Netflix (2021), 37. Where does Netflix use TCP? Netflix uses TCP for internet streaming to send packets of data for video. Additionally, Netflix specifically looks at the number of TCP connections to determine internet speeds in accordance with testing of OCAs. In another example, "After **the fill master OCA** has completed its S3 download, **it reports back to the control plane** that it now has the title stored." See https://netflixtechblog.com/netflix-and-fill-c43a32b490c0
[17a]	A network terminal to operate as a relay server in a data communication network, the data communication network including:	Netflix uses a system called **Open Connect** to deliver Netflix TV shows and movies to members world-wide. The building blocks of Open Connect are our suite of purpose-built **server appliances, called Open Connect Appliances (OCAs)**. *See* Open Connect Overview, p. 2. These are deployed directly inside ISP networks. Netflix provides the server hardware. The OCAs report to a Open Connect control plane (CCS server) to control fill behavior (adding new files to OCAs nightly) and to compute and/or store data. *See id.* p. 3-4. Accordingly, OCAs include both an input mechanism and display mechanism.

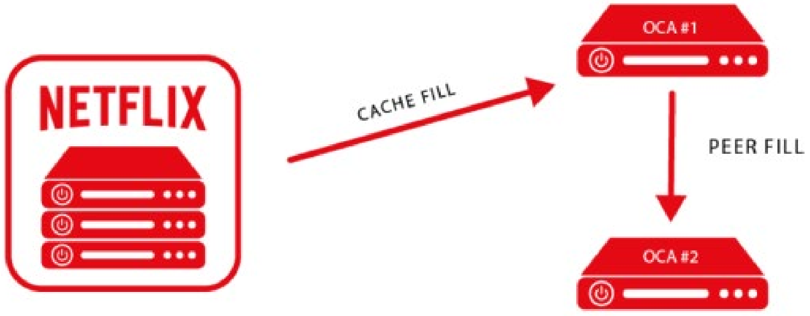
		 Client Devices 6 OCA serves files to Client Device OCAs 5 Client Device requests files from OCA 2 "Play" request 1 Reports health status, learned routes, and available files 3 Determines required files 4 Picks OCAs, sends URLs to Client Device Playback Apps Steering Service (CODA) Cache Control Service (CCS) Netflix in AWS Open Connect Appliances can be embedded in your ISP network. Embedded OCAs have the same capabilities as the OCAs that we use in our 60+ global data centers, and they are provided to qualifying ISP partners at no charge. Each embedded OCA deployment will offload a substantial amount of Netflix content traffic from peering or transport circuits. Multiple physical deployments can be distributed or clustered on a geographic or network basis to maximize local offload. Source: https://openconnect.netflix.com/en/#sample-architectures
[17b]	a plurality of terminals, a network information database, a first server from which data may be retrieved by at least one target terminal from among said	Netflix runs the operation of Open Connect from a Netflix application (CCS server) that is hosted in AWS. *See* Open Connect Overview, p. 4-5. In its global network, Netflix provides **data centers** such as an "S3" server **("first server")** housing **content ("data") on at lease one server**, and provides OCA users (such as ISP's) direct access to these data centers over the Internet that are housing the content. One or more of these **data centers** house a "**first server**" according to the claims. In deployment of Open Connect, Netflix provides Internet Service Providers with an OCA appliance direct "settlement-free interconnection (SFI)." The terminal OCA can "Connect via direct

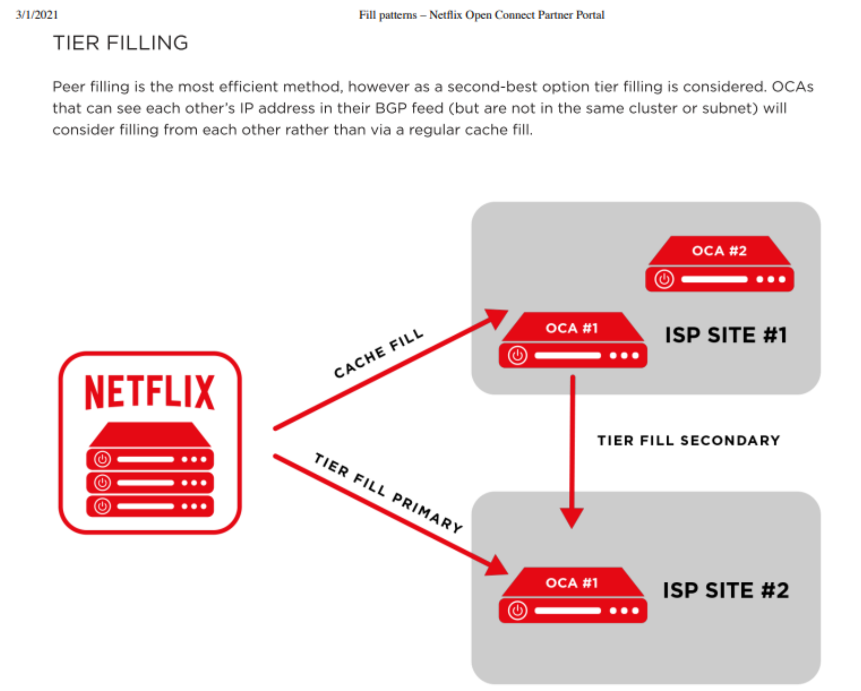
	plurality of terminals; and	Private Network Interconnect (PNI) or IXP-based SFI peering to Netflix Open Connect Appliances **in our data centers**." "Netflix has the ability to interconnect at a number of global data center facilities and public Internet Exchange fabrics as listed on our Peering Locations page. We openly peer with any network at IXP locations where we are mutually present and we consider private interconnection as appropriate." ISPs who do not currently participate in pubic peering might want to consider that a single IX port can support multiple peering sessions, providing **direct access to various content,** cloud, and network providers. Welcome to Open Connect, p. 3. Dkt 39 at p. 48. The following diagram also illustrates access from a target terminal (OCA #1, OCA #2) to a Netflix first server in "our data centers." OCAs in a cluster and on the same subnet can attempt peer filling from each other. There is also Tier filling where if in different ISP sites. deploymentguide.pdf (netflix.com)

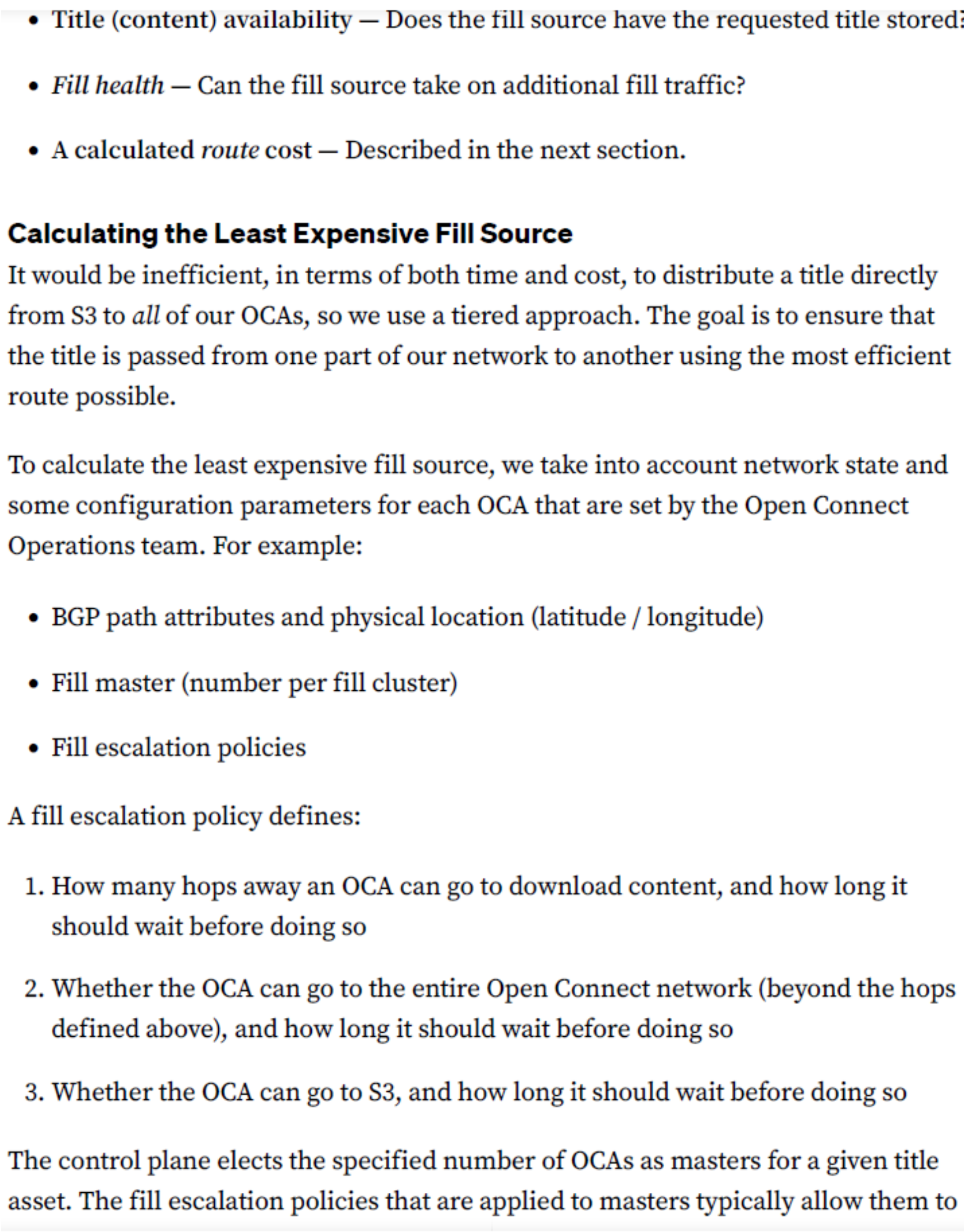
		 https://openconnect.zendesk.com/hc/en-us/articles/360035618071 2/4

		 3/1/2021 Fill patterns – Netflix Open Connect Partner Portal TIER FILLING Peer filling is the most efficient method, however as a second-best option tier filling is considered. OCAs that can see each other's IP address in their BGP feed (but are not in the same cluster or subnet) will consider filling from each other rather than via a regular cache fill. NETFLIX CACHE FILL TIER FILL PRIMARY TIER FILL SECONDARY OCA #2 OCA #1 ISP SITE #1 OCA #1 ISP SITE #2 The "target terminal selected form said plurality of terminals" language of the claim is infringed by the health and performance monitoring and OCA target terminal selection process described below in Sec. [1e].
[17c]	a main server adapted to manage selective retrieval of data from the first server by at least one target terminal selected from said plurality of terminals based	Netflix runs the operation of Open Connect from a Netflix application (main server) that is hosted in AWS. *See* Open Connect Overview, p. 4-5. In its global network, Netflix provides **data centers ("first server")** housing **content ("data") on at lease one server**, and provides OCA users (such as ISP's) direct access to these data centers over the Internet that are housing the content. One or more of these **data centers** house a "**first server**" according to the claims. In deployment of Open Connect, Netflix provides Internet Service Providers with an OCA appliance direct "settlement-free interconnection (SFI)." The terminal OCA can "Connect via direct

	on terminal performance data stored in the network information database, and wherein the main server is further adapted to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis of their relative response times;	Private Network Interconnect (PNI) or IXP-based SFI peering to Netflix Open Connect Appliances **in our data centers**." "Netflix has the ability to interconnect at a number of global data center facilities and public Internet Exchange fabrics as listed on our Peering Locations page. We openly peer with any network at IXP locations where we are mutually present and we consider private interconnection as appropriate." ISPs who do not currently participate in pubic peering might want to consider that a single IX port can support multiple peering sessions, providing **direct access to various content,** cloud, and network providers. Welcome to Open Connect, p. 3. Dkt 39 at p. 48. The following diagram also illustrates access from a target terminal (OCA #1, OCA #2) to a Netflix first server in "our data centers."

		OCAs in a cluster and on the same subnet can attempt peer filling from each other. There is also Tier filling where if in different ISP sites. deploymentguide.pdf (netflix.com)  https://openconnect.zendesk.com/hc/en-us/articles/360035618071 2/4

		 3/1/2021 Fill patterns – Netflix Open Connect Partner Portal TIER FILLING Peer filling is the most efficient method, however as a second-best option tier filling is considered. OCAs that can see each other's IP address in their BGP feed (but are not in the same cluster or subnet) will consider filling from each other rather than via a regular cache fill. NETFLIX CACHE FILL TIER FILL PRIMARY OCA #2 OCA #1 ISP SITE #1 TIER FILL SECONDARY OCA #1 ISP SITE #2 The "target terminal selected form said plurality of terminals" language of the claim is infringed by the health monitoring and OCA target terminal selection process described below in Sec. [1e]. In another example, https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states that there are "**The control plane elects the specified number of OCAs as masters. . . .**" OCAs can act as relay servers, or "masters" that target terminals can use to gain, or fill, content:



- Title (content) availability — Does the fill source have the requested title stored?
- *Fill health* — Can the fill source take on additional fill traffic?
- A calculated *route* cost — Described in the next section.

Calculating the Least Expensive Fill Source

It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to *all* of our OCAs, so we use a tiered approach. The goal is to ensure that the title is passed from one part of our network to another using the most efficient route possible.

To calculate the least expensive fill source, we take into account network state and some configuration parameters for each OCA that are set by the Open Connect Operations team. For example:

- BGP path attributes and physical location (latitude / longitude)
- Fill master (number per fill cluster)
- Fill escalation policies

A fill escalation policy defines:

1. How many hops away an OCA can go to download content, and how long it should wait before doing so
2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so
3. Whether the OCA can go to S3, and how long it should wait before doing so

The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to

[17d]	said network terminal being adapted to act as relay server for serving data retrieved from said first server to at least one target terminal by receiving and responding to transport requests sent to said network terminal, each such transport request including details of data to be retrieved, the address of the first server from which the data is to be requested by the network terminal, the addresses of at least one second target terminal to which the data retrieved from the first server is to be relayed by the network terminal and an indication of a relative	All OCA deployments are constantly monitored to ensure reliability and efficiency. Netflix **makes use of non-peak bandwidth to download the vast majority of content updates to the OCAs in network during these configurable time windows. OCAs can also download updates from each other – minimizing significant usage of internet "backbone" capacity during the update cycle.** A "desired manifest" and when needed an "emergency manifest" are posted by Netflix to the CCS server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. A "desired manifest" and when needed an "emergency manifest" are posted by Netflix to the CCS (Control Content Service) server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. Equivalently, these manifests are fetched by regularly from the CCS by each OCA terminal, which traverses the desired manifest or emergency manifest. The CCS server acts as an email inbox for manifests sent directly to the OCAs, where each OCA terminal is mandated to check regularly for manifests and download the manifests to the local OCA terminal hard drive. This desired manifest and emergency manifest and the download location data on the CCS are equivalent the "server is adapted to send transport requests direct to at least one first target terminal." The functions are substantially the same - to send a request from a main server (CCS) to a terminal (OCA) with instructions to download or "fill" content. The way this is accomplished is substantially the same. The claim states that the main server is adapted to send transport requests direct to at least one first target terminal over a network, and on the Open Connect Network, Netflix posts a desired manifest or emergency manifest to the CCS server that is intended for each OCA in the network. The results are substantially the same - to transmit commands from the main server/CCS server to a terminal/OCA to download content data in the manifest that needs to be retrieved. A "desired manifest" and when needed an "emergency manifest" are transport requests posted by Netflix to the CCS (Control Content Service) server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak

	performance of a further target terminal based on the terminal performance stored in the network information database;	hours download window, or urgently if there is an emergent need. these manifests are fetched by regularly from the CCS by each OCA terminal, which traverses the desired manifest or emergency manifest. OCAs then then query the CCS for location information files that list where each title on the desired manifest that is needed by an OCA can be downloaded. https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states a master OCA and at least one second OCA can be selected based on their relative performance: The CCS server information location list provides the address of a first server, called the "S3" server, for a download location and provides other download locations of a second and additional master OCAs: It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to all of our OCAs, so we use a tiered approach. . . . A fill escalation policy defines: 1. How many hops away an OCA can go to download content, and how long it should wait before doing so 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so 3. Whether the OCA can go to S3, and how long it should wait before doing so The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to reach farther with less delay in order to grab that content and then share it locally with non-masters. As stated in [1e], the CCS monitors the "fill health" and performance of download locations, which is based on performance of the OCAs, to determine if that OCA will be selected as a download location or not. This desired manifest and emergency manifest along with the downloaded location information files and fill policy for master OCAs and a second (target terminal) OCA to fill from a (first terminal) OCA on the CCS are equivalent the "each such transport request includes details of data to be retrieved, the address of the first server from which the data is to be requested by the first

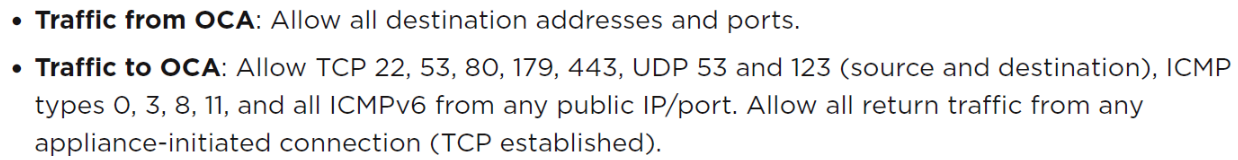
		target terminal, the addresses of at least one second target terminal to which the data from the first server to be relayed by the first target terminal." The functions are substantially the same - to send a request from a main server (CCS) to a terminal (OCA) with instructions to download or "fill" content from specific master terminal/OCA addresses and includes an address of at least one second terminal/OCA. The way this is accomplished is substantially the same. The claim states that the main server is adapted to send transport requests that include download locations direct to at least one first target terminal over a network, and on the Open Connect Network, Netflix posts a desired manifest or emergency manifest to the CCS server along with download locations that are intended for each OCA to read on a regular basis. The results are substantially the same - to transmit commands from the main server/CCS server to a terminal/OCA to download content data in the manifest that needs to be retrieved from various addresses. The control plane elects the specified number of OCAs as masters for a given title asset. https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 Thus, the CCS server selects OCA terminals as a download location for a given title asset. The main server on CCS is further adapted to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis of their relative response times. At https://netflixtechblog.com/netflix-and-fill-c43a32b490c0, Netflix wrote: The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title. The determination of the list takes into consideration several high-level factors: Title (content) availability — Does the fill source have the requested title stored? **Fill health — Can the fill source take on additional fill traffic?** A calculated route cost — Described in the next section. A fill escalation policy defines:

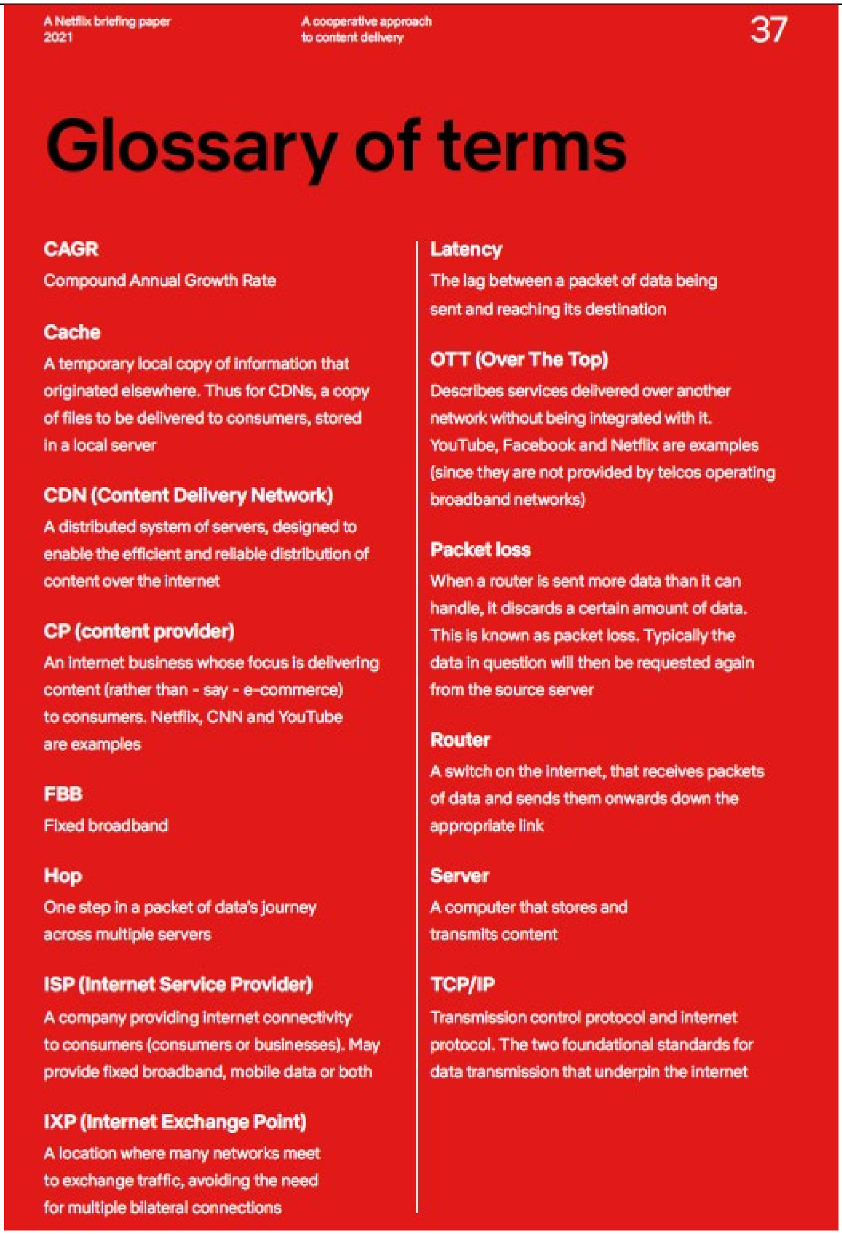
		13. How many hops away an OCA can go to download content, and **how long it should wait before doing so** 14. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), **and how long it should wait before doing so** 15. Whether the OCA can go to S3, **and how long it should wait before doing so** (Emphasis added.) The CCS server monitors the "fill health" of each potential download locations, or fill sources that includes OCAs, and uses a fill escalation policy based upon response times to determine if the performance of the download location can take on additional traffic or not and how long the OCA should wait for a response before escalation to the next download locations. Thus, the manifest is assembled by the CCS server and download locations are assembled based on OCA performance information, and the manifest is sent to a given OCA. Therefore, the transport request is sent on a basis of said terminal performance information.
[17e]	wherein said network terminal adapted to act as relay server are further adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein	OCAs can also download updates from each other – minimizing significant usage of internet "backbone" capacity during the update cycle. The OCAs work in a network to distribute updates among each other and to include further OCAs to which updates and content can be sent. *See* Open Connect Overview, p. 5; Fill Patterns, pp. 1-3. Netflix' OCA that are adapted to act as relay servers (see 1d above) are adapted to modify transport requests received from the main server or from other relay servers and transmit the modified transport request to selected target terminals that includes addresses of further target terminals. The CCS server will to order OCA terminals to peer or tier fill from using OCAs selected by the CCS server. The CCS server uses Appliance Section Criteria to select OCA terminals as targets in the to transfer titles from: 1) the terminal appliance that receives the most-specific route; 2) the terminal appliance that receives the route to the client's netblock with the shortest AS path; 3) the terminal appliance that receives the route to the client's netblock with the lowest multi-exit discriminator; 4) the geographically closest appliance. The CCS server includes the URL addresses of these master or target terminals in the desired manifest, which is loaded by an OCA terminal in its memory or hard drive space in order to select an OCA for downloading titles from.

	the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server; and	After receiving the desired manifest, an OCA terminal will compare the manifest with its own “actual manifest”, or the list of titles currently residing with the OCA terminal. The titles missing from the actual manifest are termed the “delta” or difference between actual and desired manifest. The OCA terminal will then query the CCS terminal for a list of download locations for each title on the delta. The CCS responds, as stated in [1f] with a list of URLs that are downloadable locations of master OCAs for each individual title needed by an OCA to fill its delta: “OCAs communicate at regular intervals with the control plane services, requesting (among other things) a manifest file that contains the list of titles they should be storing and serving to members. If there is a **delta** between the list of titles in the manifest and what they are currently storing, each OCA will send a request, during its configured fill window, **that includes a list of the new or updated titles that it needs**. The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title.” (Emphasis added.) *See* https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 The action of an OCA requesting download locations (master OCAs) for its delta list from the CCS, and then requesting a delta-listed title from the list of a master OCAs, is equivalent to a modified transport request. The OCA is using a modified list of titles (delta or missing titles list from the desired manifest) to request a title or titles from further target terminals, or master OCAs. This is equivalent to: “terminals adapted to act as relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server.” The function of the two actions are substantially the same, which is to transmit to other selected relay servers or OCAs a modification of the original transport request, or desired manifest, along with URLs of those relay servers or URLs of master OCAs. The way the actions are performed are substantially the same. A delta is a modification of the manifest list, or in other words a subset list of what it is supposed to download. Instead of the terminal, or OCA, transmitting the delta list to another OCA, an “actual manifest” is sent to the CCS server which responds to the OCA with a list of URL locations to download the modified list titles of its delta list, after which the delta list is then sent to another OCA via a series of modified transfer requests. The result of these actions are

		substantially similar: further relay terminal addresses are sent to the OCAs, a modification of the original transport request (desired manifest) is transmitted to selected relay terminals (single or multiple delta titles to selected OCAs) are sent to addresses (URLs) of further selected relay servers (OCAs) in the form of a request for one or more titles from one or more master OCAs. Alternatively and equivalently, Netflix documentation discloses that OCA terminals, if they are clustered or if they are in the same subnet, will attempt to peer or tier fill from each other. https://openconnect.zendesk.com/hc/en-us/articles/360035618071-Fill-patterns OCA terminals in a subnet or cluster broadcast their IP and physical locations to one another and save this information. In general, appliances determine where to receive fill using selection criteria that is used by Netflix client devices. The OCA terminals then use a similar Appliance Section Criteria as the CCS server uses to select OCA terminals as targets in the subnet or cluster to transfer titles from: 1) the terminal appliance that receives the most-specific route; 2) the appliance terminal that receives the route to the client's netblock with the shortest AS path; 3) the appliance terminal that receives the route to the client's netblock with the lowest multi-exit discriminator; 4) the geographically closest terminal. The OCA terminal includes the URL addresses of these terminals in its memory or hard drive space in order to select an OCA for downloading titles from. See *Fill Patterns*, pp. 1-4. After receiving the desired manifest, an OCA terminal will compare the manifest with its own "actual manifest", or the list of titles currently residing with the OCA terminal. The titles missing from the actual manifest are termed the "delta" or difference between actual and desired manifest. After selecting an OCA master using the selection criteria, the OCA terminal will transmit and request the delta list items to the selected OCA(s) in the subnet or cluster in the form of download requests for each title using the URL of the target OCA. This is equivalent to: "terminals adapted to act as relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server."

		The function of the two actions are substantially the same, which is to transmit to other selected relay servers or OCAs a modified list of the original transport request or manifest. The way the actions are performed are substantially the same. A modified desired manifest list is a subset list called the "delta." A relay server (OCA) creates a modified transport request (delta list) of its missing titles and transports this list to another relay server (OCA sends requests to other selected OCAs for the titles on the delta list in a series of requests) and includes addresses of the other relay servers (URLs of the selected OCA terminals in the request as an address to the selected OCAs). The result of these actions are substantially similar: a modification of the original transport request (desired manifest) is transmitted to selected relay terminals (selected OCAs) with addresses (URLs) of the selected relay servers (OCAs).
[17f]	wherein data to be retrieved by said target terminals are divided into a series of packets for transmission to said target terminals and each of said terminals are adapted to communicate directly with said main server to acknowledge receipt of the last packet of a series routed thereto.	Netflix' OCAs are adapted to communicate with the main server which is hosted by AWS. The OCA's "Report their status to the Open Connect control panel services in Amazon Web Services": The Open Connect network can make parallelized cache fill transfers. As such, it must be transferring "packets" of the file. There is evidence that each file is downloaded for cache fill in parallel in packet sizes up to 16 kilobytes. Additionally, the last packet of a download for a cache fill is associated with a notice to the CCS that the download has been completed. The system can also use parallel processes to perform cache transfer using TCP/IP protocols. This is only possible if the CCS is working on a different section of the file at the same time, which is packetized file transfer. There is also evidence that the cache fill transfer uses TCP/IP protocols for packetized data transfer. TCP data communication protocol, which most of the Internet included Netflix uses, requires an acknowledgement of packets when content is transmitted: TCP is a reliable byte stream delivery service which guarantees that all bytes received will be identical and in the same order as those sent. Since packet transfer by many networks is not reliable, **TCP achieves this using a technique known as positive acknowledgement with re-transmission**. This **requires the receiver to respond with an acknowledgement message as it receives the data**. The sender keeps a record of each packet it sends and maintains a timer from

		when the packet was sent. The sender re-transmits a packet if the timer expires before receiving the acknowledgement. The timer is needed in case a packet gets lost or corrupted.[Comer, Douglas E. (2006). Internetworking with TCP/IP: Principles, Protocols, and Architecture. Vol. 1 (5th ed.). Prentice Hall. ISBN 978-0-13-187671-2 Netflix requires network traffic to OCA be in TCP protocol:  • **Traffic from OCA**: Allow all destination addresses and ports. • **Traffic to OCA**: Allow TCP 22, 53, 80, 179, 443, UDP 53 and 123 (source and destination), ICMP types 0, 3, 8, 11, and all ICMPv6 from any public IP/port. Allow all return traffic from any appliance-initiated connection (TCP established). https://openconnect.zendesk.com/hc/en-us/articles/360035533071#routi



Glossary of terms

CAGR
Compound Annual Growth Rate

Cache
A temporary local copy of information that originated elsewhere. Thus for CDNs, a copy of files to be delivered to consumers, stored in a local server

CDN (Content Delivery Network)
A distributed system of servers, designed to enable the efficient and reliable distribution of content over the internet

CP (content provider)
An internet business whose focus is delivering content (rather than - say - e-commerce) to consumers. Netflix, CNN and YouTube are examples

FBB
Fixed broadband

Hop
One step in a packet of data's journey across multiple servers

ISP (Internet Service Provider)
A company providing internet connectivity to consumers (consumers or businesses). May provide fixed broadband, mobile data or both

IXP (Internet Exchange Point)
A location where many networks meet to exchange traffic, avoiding the need for multiple bilateral connections

Latency
The lag between a packet of data being sent and reaching its destination

OTT (Over The Top)
Describes services delivered over another network without being integrated with it. YouTube, Facebook and Netflix are examples (since they are not provided by telcos operating broadband networks)

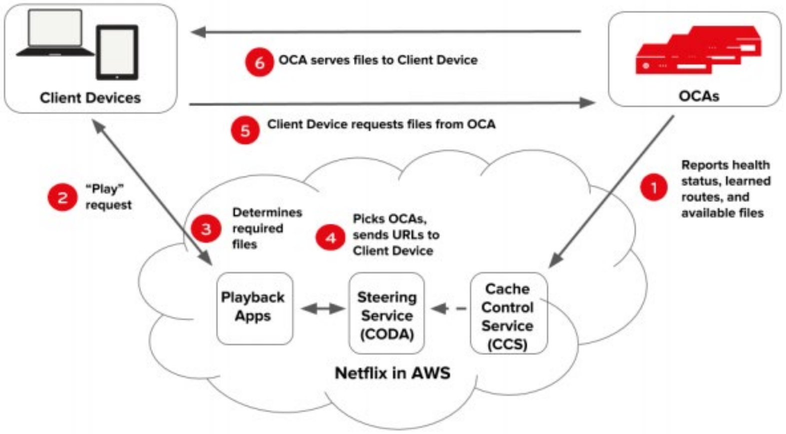
Packet loss
When a router is sent more data than it can handle, it discards a certain amount of data. This is known as packet loss. Typically the data in question will then be requested again from the source server

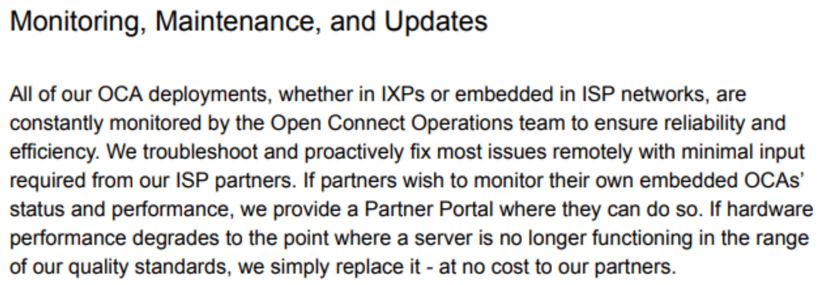
Router
A switch on the internet, that receives packets of data and sends them onwards down the appropriate link

Server
A computer that stores and transmits content

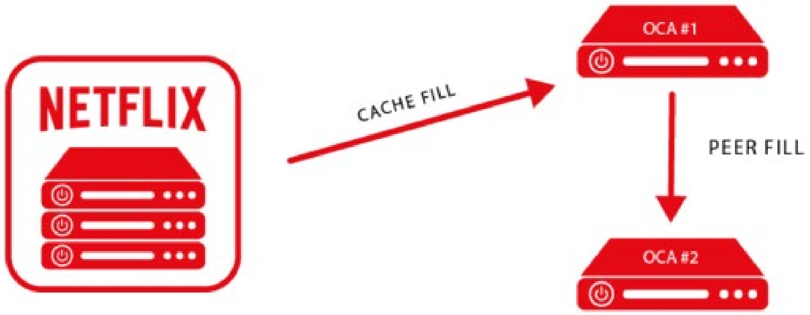
TCP/IP
Transmission control protocol and internet protocol. The two foundational standards for data transmission that underpin the internet

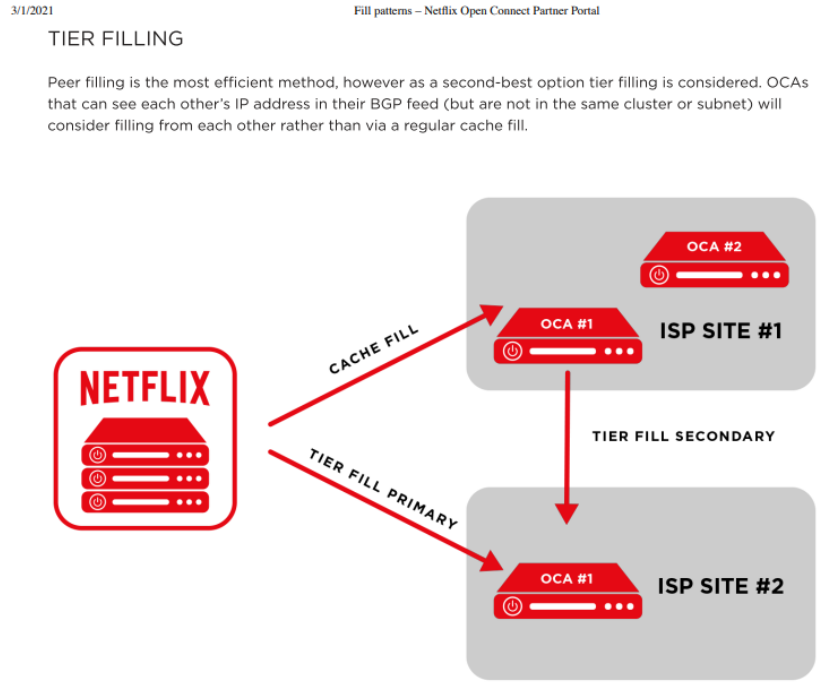
		“A cooperative approach to content delivery,” Netflix (2021), 37. Where does Netflix use TCP? Netflix uses TCP for internet streaming to send packets of data for video. Additionally, Netflix specifically looks at the number of TCP connections to determine internet speeds in accordance with testing of OCAs. In another example, “After **the fill master OCA** has completed its S3 download, **it reports back to the control plane** that it now has the title stored.” See https://netflixtechblog.com/netflix-and-fill-c43a32b490c0
[19a]	A computer program product for enabling a network server to operate as a main server in a data communication network, the data communication network including:	Netflix uses a system called **Open Connect** to deliver Netflix TV shows and movies to members world-wide. The building blocks of Open Connect are our suite of purpose-built **server appliances, called Open Connect Appliances (OCAs)**. *See* Open Connect Overview, p. 2. These are deployed directly inside ISP networks. Netflix provides the server hardware. The OCAs report to a Open Connect control plane to control fill behavior (adding new files to OCAs nightly) and to compute and/or store data. *See id.* p. 3-4. Accordingly, OCAs include both an input mechanism and display mechanism.. Netflix runs the operation of Open Connect from a Netflix application (CCS server) that is hosted in AWS. It is on information and belief that because Netflix Open Connect system includes the infringing elements and performs functions that infringe, there is the requisite software, or computer program, to carry out the functionality of Netflix’s Open Connect system.

		 Client Devices OCAs 6 OCA serves files to Client Device 5 Client Device requests files from OCA 2 "Play" request 1 Reports health status, learned routes, and available files 3 Determines required files 4 Picks OCAs, sends URLs to Client Device Playback Apps Steering Service (CODA) Cache Control Service (CCS) Netflix in AWS Open Connect Appliances can be embedded in your ISP network. Embedded OCAs have the same capabilities as the OCAs that we use in our 60+ global data centers, and they are provided to qualifying ISP partners at no charge. Each embedded OCA deployment will offload a substantial amount of Netflix content traffic from peering or transport circuits. Multiple physical deployments can be distributed or clustered on a geographic or network basis to maximize local offload. Source: https://openconnect.netflix.com/en/#sample-architectures
[19b]	a plurality of terminals, a network information database and a first server which from which data be retrieved by at least one target terminal from among said plurality of terminals, at least	Netflix runs the operation of Open Connect from a Netflix application (main server) that is hosted in AWS. *See* Open Connect Overview, p. 4-5. In its global network, Netflix provides **data centers ("first server")** housing **content ("data") on at lease one server**, and provides OCA users (such as ISP's) direct access to these data centers over the Internet that are housing the content. One or more of these **data centers** house a "**first server**" according to the claims. In deployment of Open Connect, Netflix provides Internet Service Providers with an OCA appliance direct "settlement-free interconnection (SFI)." The terminal OCA can "Connect via direct Private Network Interconnect (PNI) or IXP-based SFI peering to Netflix Open Connect Appliances **in our data centers**."

	two of said terminals being adapted to act as relay servers for serving data retrieved from said first server to at least one further target terminal based on terminal performance information stored in the network information database, said main server being distinct from said first server, said computer program product comprising:	“Netflix has the ability to interconnect at a number of global data center facilities and public Internet Exchange fabrics as listed on our Peering Locations page. We openly peer with any network at IXP locations where we are mutually present and we consider private interconnection as appropriate.” ISPs who do not currently participate in pubic peering might want to consider that a single IX port can support multiple peering sessions, providing **direct access to various content,** cloud, and network providers. Welcome to Open Connect, p. 3. Dkt 39 at p. 48. The following diagram also illustrates access from a target terminal (OCA #1, OCA #2) to a Netflix first server in “our data centers.” Netflix constantly measures and analyzes [OCA] performance and augment capacity as requirements evolve. *See* Open Connect Overview, p. 3. All OCA deployments are constantly monitored by the Open Connect Operations team to ensure reliability and efficiency. We troubleshoot and proactively fix most issues remotely with minimal input required from our ISP partners. *See* Open Connect Overview, p. 5. Additionally, OCAs periodically report health. *Id.* at 4.  Monitoring, Maintenance, and Updates All of our OCA deployments, whether in IXPs or embedded in ISP networks, are constantly monitored by the Open Connect Operations team to ensure reliability and efficiency. We troubleshoot and proactively fix most issues remotely with minimal input required from our ISP partners. If partners wish to monitor their own embedded OCAs’ status and performance, we provide a Partner Portal where they can do so. If hardware performance degrades to the point where a server is no longer functioning in the range of our quality standards, we simply replace it - at no cost to our partners. The main server on CCS is further adapted to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis

		of their relative response times. At https://netflixtechblog.com/netflix-and-fill-c43a32b490c0, Netflix wrote: The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title. The determination of the list takes into consideration several high-level factors: Title (content) availability — Does the fill source have the requested title stored? **Fill health — Can the fill source take on additional fill traffic?** A calculated route cost — Described in the next section. A fill escalation policy defines: 1. How many hops away an OCA can go to download content, and **how long it should wait before doing so** 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), **and how long it should wait before doing so** 3. Whether the OCA can go to S3, **and how long it should wait before doing so** (Emphasis added.) The CCS server monitors the "fill health" of each potential download locations, or fill sources that includes OCAs, and uses a fill escalation policy based upon response times to determine if the performance of the download location can take on additional traffic or not and how long the OCA should wait for a response before escalation to the next download locations. Saving this information to a database is common practice within network management

		OCAs in a cluster and on the same subnet can attempt peer filling from each other. There is also Tier filling where if in different ISP sites. deploymentguide.pdf (netflix.com)  https://openconnect.zendesk.com/hc/en-us/articles/360035618071 2/4



3/1/2021 Fill patterns – Netflix Open Connect Partner Portal

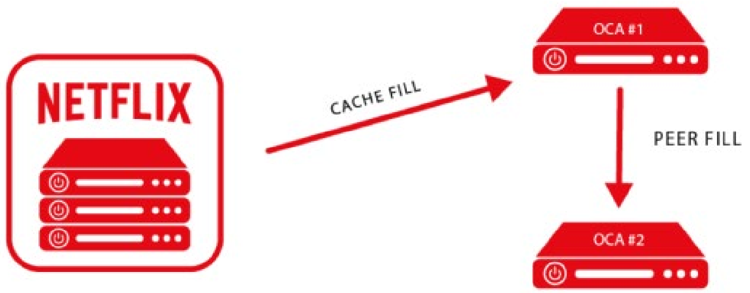
TIER FILLING

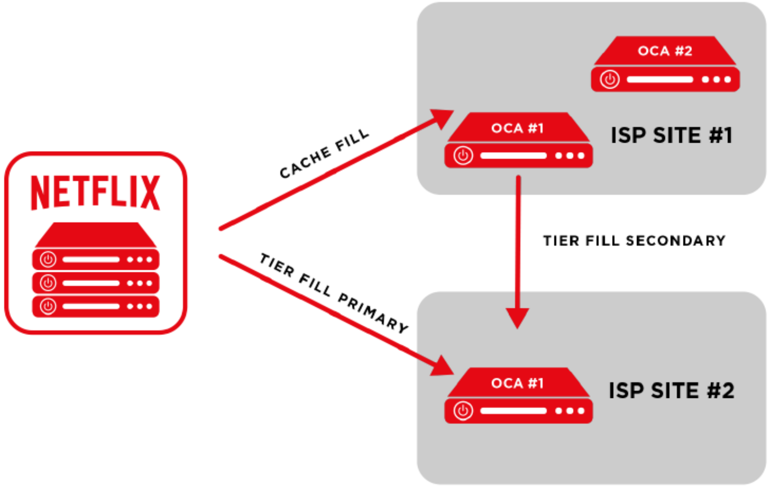
Peer filling is the most efficient method, however as a second-best option tier filling is considered. OCAs that can see each other's IP address in their BGP feed (but are not in the same cluster or subnet) will consider filling from each other rather than via a regular cache fill.

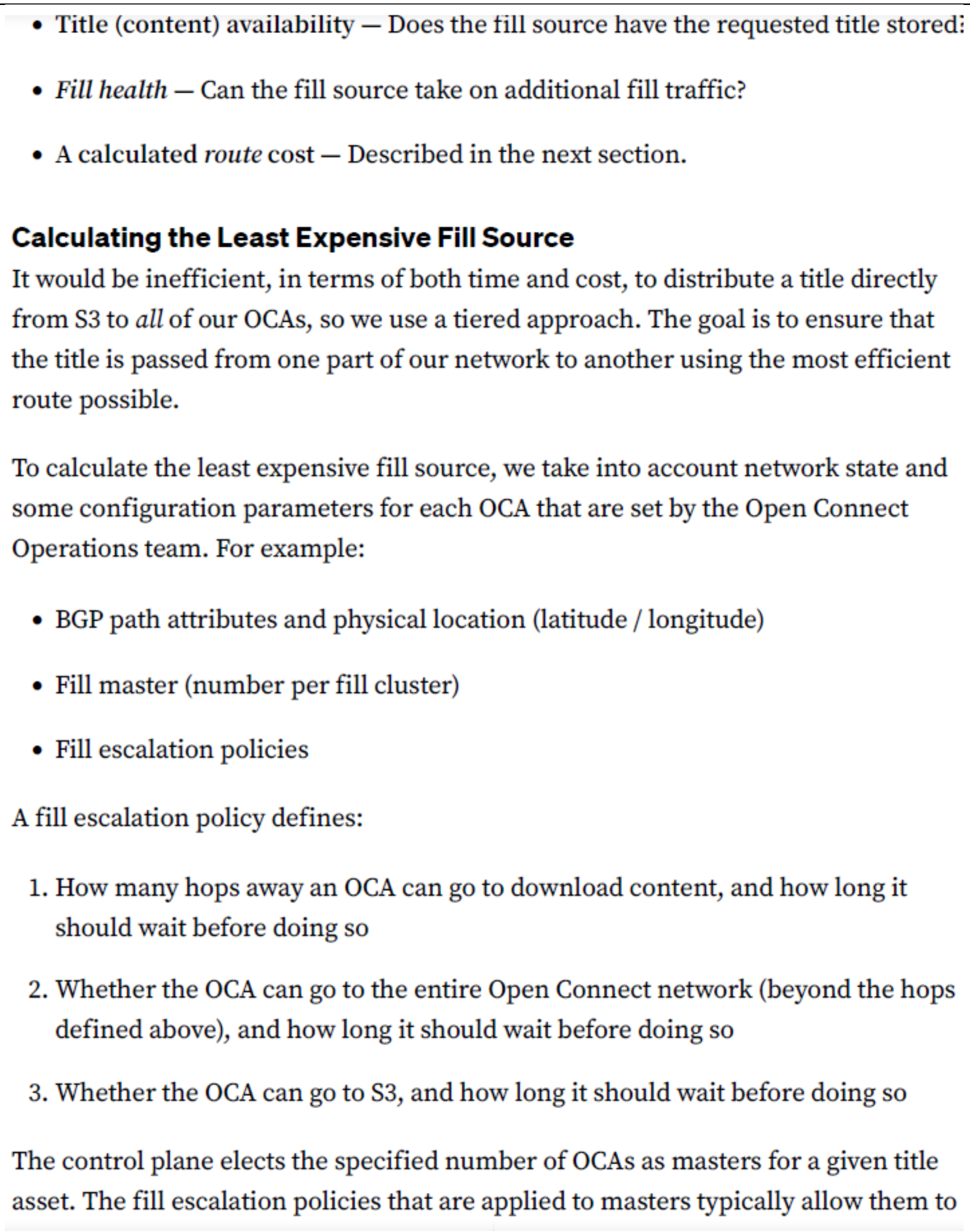
The "target terminal selected form said plurality of terminals" language of the claim is infringed by the health monitoring and OCA target terminal selection process described below in Sec. [1e].

A "first server" is identified above in Sec. [1a] as a server within Netflix' data centers.

According to Netflix' network architecture, OCAs in a cluster and on the same subnet can attempt peer filling content "cache fill" from the first server (data center) to OCA #1. OCA#1 then acts as a "relay server" to fill cached content to each other, from OCA#1 to OCA#2. OCA#2 becomes a "target terminal" in this example that is served data from the relay server (OCA#1) retrieved from the first server (data center). See https://openconnect.netflix.com/deploymentguide.pdf.

		 https://openconnect.zendesk.com/hc/en-us/articles/360035618071 2/4 Netflix's first server (data center) can cache fill content to a terminal OCA#1, which then acts as a "relay server" to fill a target terminal (OCA#2) using "tier filling." "Peer filling is the most efficient method, however as a second-best option tier filling is considered. OCAs that can see each other's IP address in their BGP feed (but are not in the same cluster or subnet) will consider filling from each other rather than via a regular cache fill."

		 NETFLIX CACHE FILL TIER FILL PRIMARY TIER FILL SECONDARY OCA #2 OCA #1 ISP SITE #1 OCA #1 ISP SITE #2 See https://openconnect.netflix.com/deploymentguide.pdf. In another example, https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states that there are "**The control plane elects the specified number of OCAs as masters. . . .**" OCAs can act as relay servers, or "masters" that target terminals can use to gain, or fill, content:



- Title (content) availability — Does the fill source have the requested title stored?
- *Fill health* — Can the fill source take on additional fill traffic?
- A calculated *route* cost — Described in the next section.

Calculating the Least Expensive Fill Source

It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to *all* of our OCAs, so we use a tiered approach. The goal is to ensure that the title is passed from one part of our network to another using the most efficient route possible.

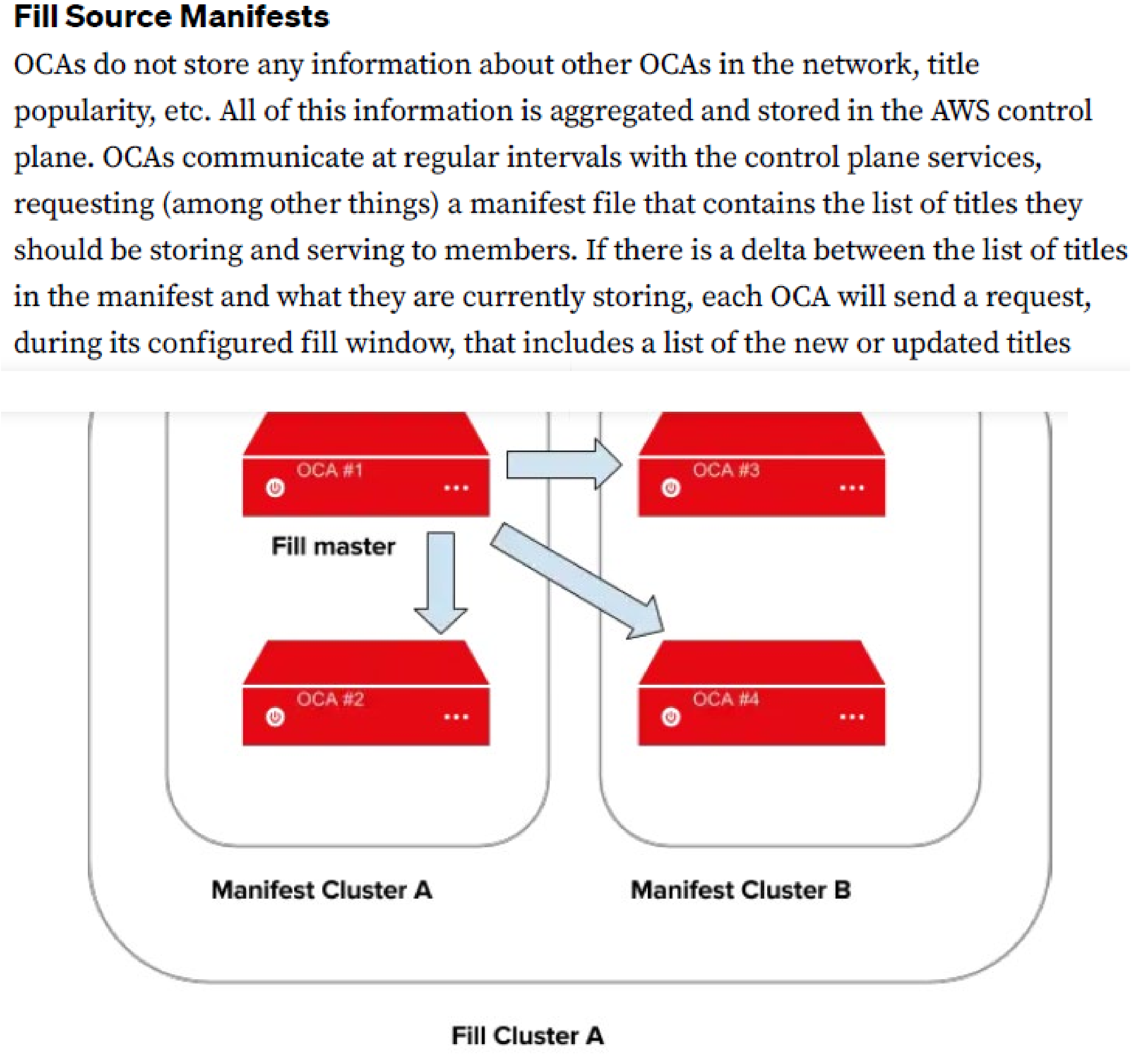
To calculate the least expensive fill source, we take into account network state and some configuration parameters for each OCA that are set by the Open Connect Operations team. For example:

- BGP path attributes and physical location (latitude / longitude)
- Fill master (number per fill cluster)
- Fill escalation policies

A fill escalation policy defines:

1. How many hops away an OCA can go to download content, and how long it should wait before doing so
2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so
3. Whether the OCA can go to S3, and how long it should wait before doing so

The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to

		 **Fill Source Manifests** OCAs do not store any information about other OCAs in the network, title popularity, etc. All of this information is aggregated and stored in the AWS control plane. OCAs communicate at regular intervals with the control plane services, requesting (among other things) a manifest file that contains the list of titles they should be storing and serving to members. If there is a delta between the list of titles in the manifest and what they are currently storing, each OCA will send a request, during its configured fill window, that includes a list of the new or updated titles OCA #1 Fill master OCA #2 OCA #3 OCA #4 Manifest Cluster A Manifest Cluster B Fill Cluster A

		When the second tier of OCAs complete their download, they report back their status, other OCAs can then fill from them, and so on. This process continues during the fill window. If there are titles being stored on an OCA that are no longer needed, they are put into a delete manifest and then deleted after a period of time that ensures we don't interrupt any live sessions. As the sun moves west and more members begin streaming, the fill window in this time zone ends, and the fill pattern continues as the fill window moves across other time zones — until enough of the OCAs in our global network that need to be able to serve this new title have it stored.

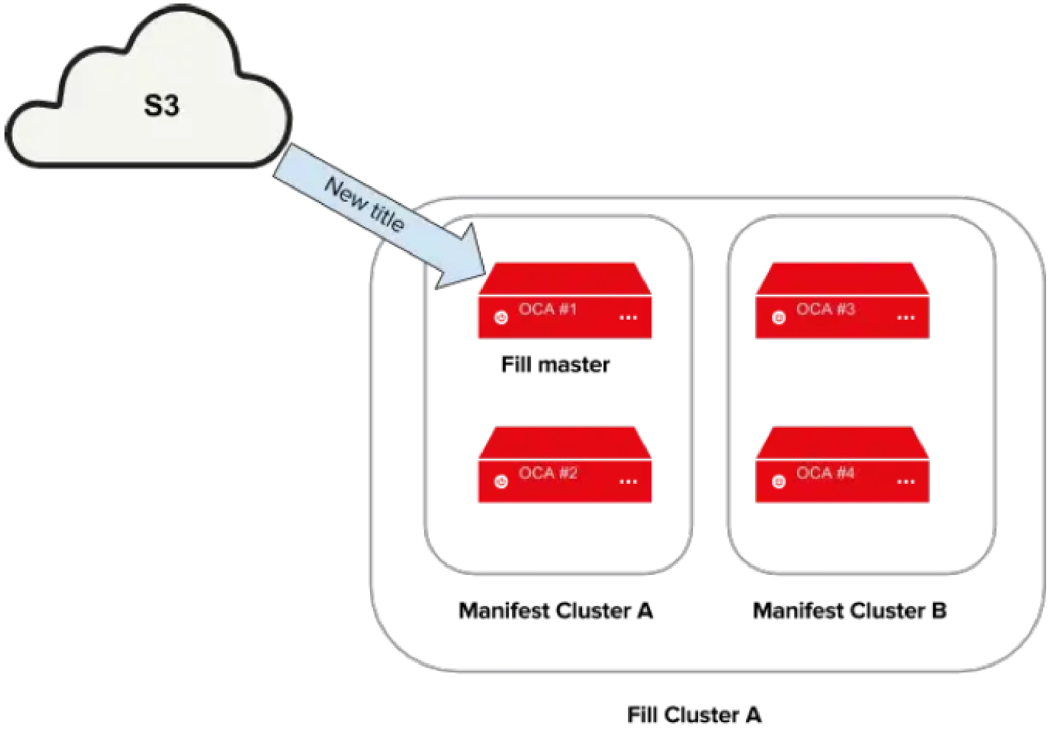
1. Peer fill: Available OCAs within the same manifest cluster or the same subnet

2. Tier fill: Available OCAs outside the manifest cluster configuration

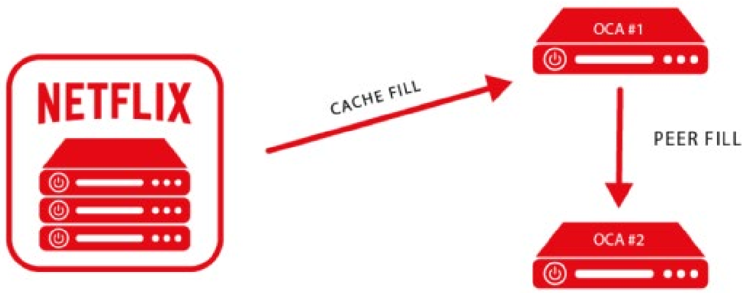
3. Cache fill: Direct download from S3

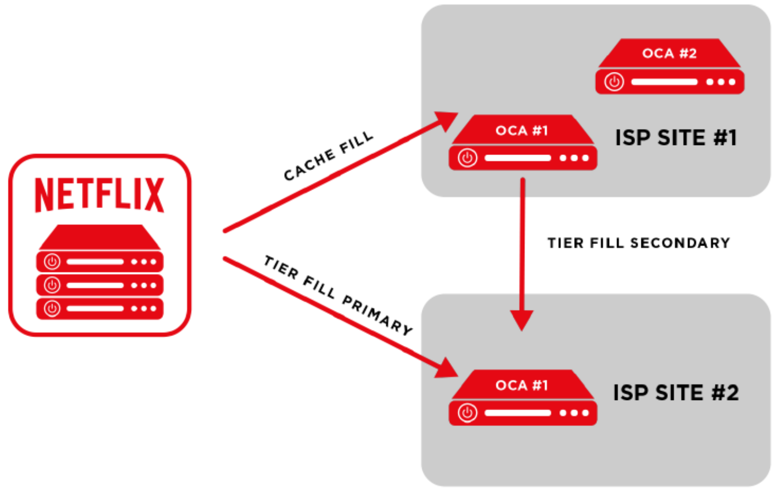
Example Scenario

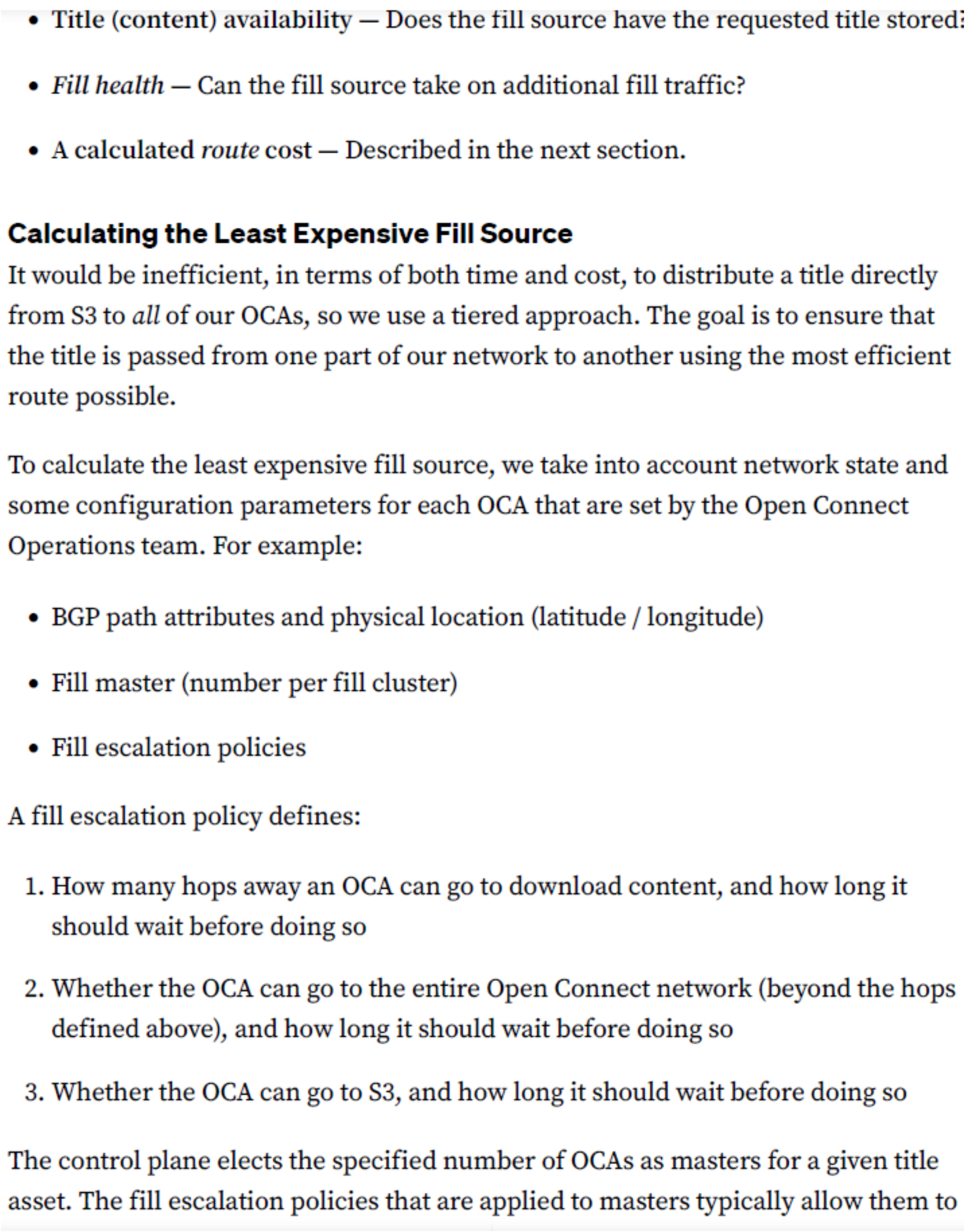
After the fill master OCA has completed its S3 download, it reports back to the control plane that it now has the title stored. The next time the other OCAs communicate with the control plane to request a fill source for this title, they are given the option to fill from the fill master.



[19c]	a non-transitory computer usable medium having computer readable program code means embodied in said non-transitory medium, said computer readable program code means including:	It is on information and belief that Netflix Open Connect system includes the requisite software, or computer program, to carry out the functionality of Netflix's Open Connect system.
[19d]	computer readable program code for causing said network server to manage selective retrieval of data from said first server by at least one target terminal selected from said plurality of terminals; and wherein said network server to monitor response times of terminals in the network and in which terminals	A "first server" or S3 is identified above in Sec. [1a] as a server within Netflix' data centers, and at least two of the OCAs are adapted to act as relay servers for serving data retrieved from said first server to at least one OCA terminal. According to Netflix' network architecture, OCAs in a cluster and on the same subnet can attempt peer filling content "cache fill" from the first server to OCA #1. OCA#1 then acts as a "relay server" to fill cached content to each other, from OCA#1 to OCA#2. OCA#2 becomes a "target terminal" in this example that is served data from the relay server (OCA#1) retrieved from the first server (data center). See https://openconnect.netflix.com/deploymentguide.pdf.

	are selected to act as relay servers for a particular data transfers on the basis of their relative response times,	 https://openconnect.zendesk.com/hc/en-us/articles/360035618071 2/4

		In another example, Netflix's first server (data center) can cache fill content to a terminal OCA#1, which then acts as a "relay server" to fill a target terminal (OCA#2) using "tier filling." "Peer filling is the most efficient method, however as a second-best option tier filling is considered. OCAs that can see each other's IP address in their BGP feed (but are not in the same cluster or subnet) will consider filling from each other rather than via a regular cache fill."  NETFLIX CACHE FILL TIER FILL PRIMARY TIER FILL SECONDARY OCA #2 OCA #1 ISP SITE #1 OCA #1 ISP SITE #2 See https://openconnect.netflix.com/deploymentguide.pdf. In another example, https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states that there are "**The control plane elects the specified number of OCAs as masters. . . .**" OCAs can act as relay servers, or "masters" that target terminals can use to gain, or fill, content:



- Title (content) availability — Does the fill source have the requested title stored?
- *Fill health* — Can the fill source take on additional fill traffic?
- A calculated *route* cost — Described in the next section.

Calculating the Least Expensive Fill Source

It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to *all* of our OCAs, so we use a tiered approach. The goal is to ensure that the title is passed from one part of our network to another using the most efficient route possible.

To calculate the least expensive fill source, we take into account network state and some configuration parameters for each OCA that are set by the Open Connect Operations team. For example:

- BGP path attributes and physical location (latitude / longitude)
- Fill master (number per fill cluster)
- Fill escalation policies

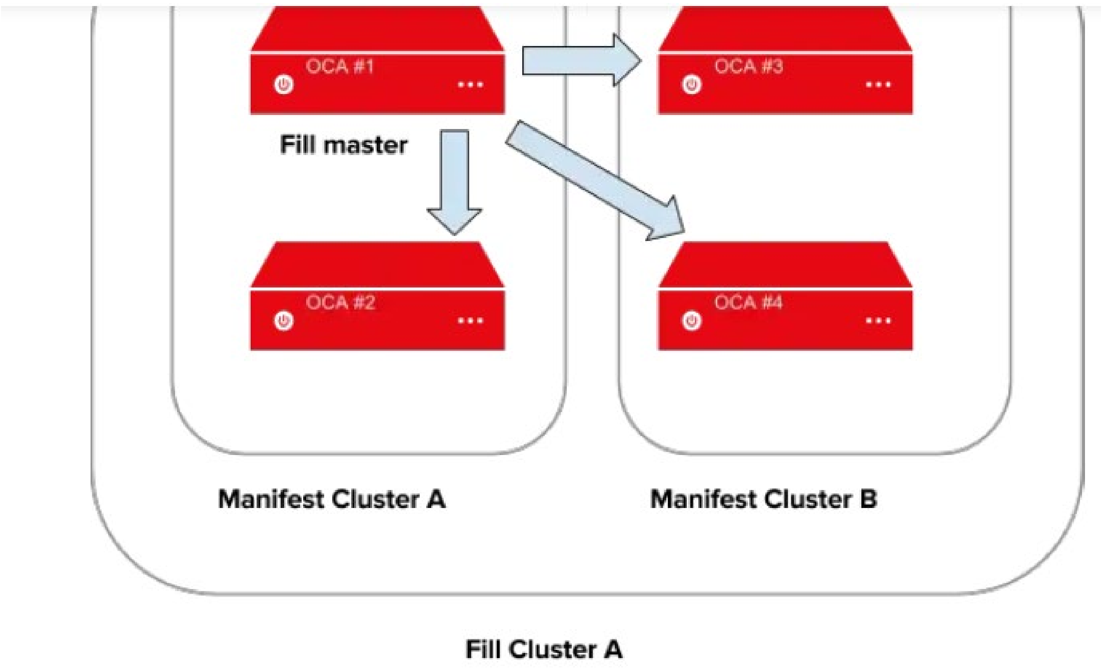
A fill escalation policy defines:

1. How many hops away an OCA can go to download content, and how long it should wait before doing so
2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so
3. Whether the OCA can go to S3, and how long it should wait before doing so

The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to

The main server on

		CCS is further adapted to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis of their relative response times. At https://netflixtechblog.com/netflix-and-fill-c43a32b490c0, Netflix wrote: The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title. The determination of the list takes into consideration several high-level factors: Title (content) availability — Does the fill source have the requested title stored? **Fill health — Can the fill source take on additional fill traffic?** A calculated route cost — Described in the next section. A fill escalation policy defines: 1. How many hops away an OCA can go to download content, and **how long it should wait before doing so** 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), **and how long it should wait before doing so** 3. Whether the OCA can go to S3, **and how long it should wait before doing so** (Emphasis added.) The CCS server monitors the "fill health" of each potential download locations, or fill sources that includes OCAs, and uses a fill escalation policy based upon response times to determine if the performance of the download location can take on additional traffic or not and how long the OCA should wait for a response before escalation to the next download locations.

		OCA #1 OCA #3 Fill master OCA #2 OCA #4 Manifest Cluster A Manifest Cluster B Fill Cluster A

		When the second tier of OCAs complete their download, they report back their status, other OCAs can then fill from them, and so on. This process continues during the fill window. If there are titles being stored on an OCA that are no longer needed, they are put into a delete manifest and then deleted after a period of time that ensures we don’t interrupt any live sessions. As the sun moves west and more members begin streaming, the fill window in this time zone ends, and the fill pattern continues as the fill window moves across other time zones — until enough of the OCAs in our global network that need to be able to serve this new title have it stored.

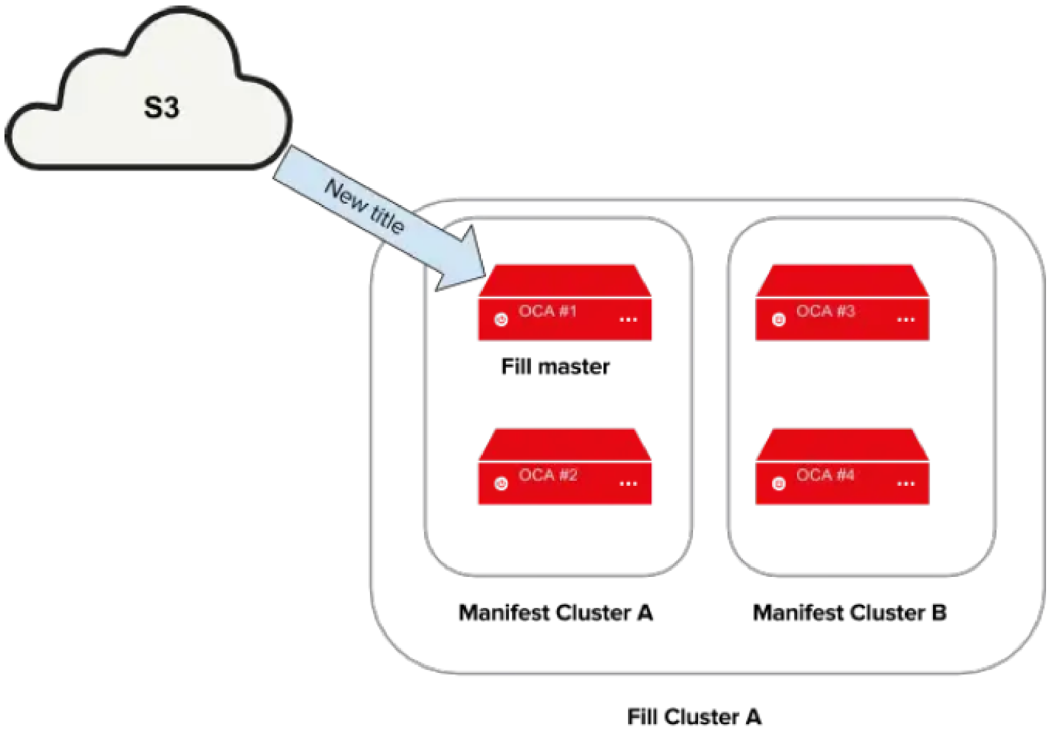
1. Peer fill: Available OCAs within the same manifest cluster or the same subnet

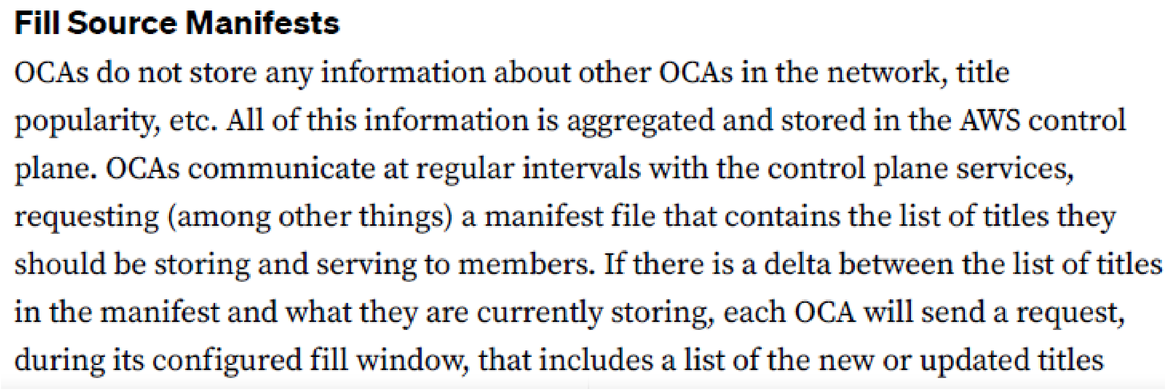
2. Tier fill: Available OCAs outside the manifest cluster configuration

3. Cache fill: Direct download from S3

Example Scenario

After the fill master OCA has completed its S3 download, it reports back to the control plane that it now has the title stored. The next time the other OCAs communicate with the control plane to request a fill source for this title, they are given the option to fill from the fill master.



[19e]	computer readable program code for causing said network server to send transport requests direct to at least one first target terminal that is adapted to act as a relay server, each such transport request including details of data to be retrieved, the address of the first server from which the data is to be requested by the first target terminal, the addresses of at least one second target terminal to which the data retrieved from the first server is to be relayed by the first target terminal and an indication of a relative performance of a further target	All OCA deployments are constantly monitored to ensure reliability and efficiency. Netflix **makes use of non-peak bandwidth to download the vast majority of content updates to the OCAs in network during these configurable time windows. OCAs can also download updates from each other – minimizing significant usage of internet "backbone" capacity during the update cycle.** A "desired manifest" and when needed an "emergency manifest" are posted by Netflix to the CCS server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need.  **Fill Source Manifests** OCAs do not store any information about other OCAs in the network, title popularity, etc. All of this information is aggregated and stored in the AWS control plane. OCAs communicate at regular intervals with the control plane services, requesting (among other things) a manifest file that contains the list of titles they should be storing and serving to members. If there is a delta between the list of titles in the manifest and what they are currently storing, each OCA will send a request, during its configured fill window, that includes a list of the new or updated titles A "desired manifest" and when needed an "emergency manifest" are posted by Netflix to the CCS (Control Content Service) server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. Equivalently, these manifests are fetched by regularly from the CCS by each OCA terminal, which traverses the desired manifest or emergency manifest. The CCS server acts as an email inbox for manifests sent directly to the OCAs, where each OCA terminal is mandated to check regularly for manifests and download the manifests to the local OCA terminal hard drive.

	terminal based on the terminal performance information stored in the network information database;	This desired manifest and emergency manifest and the download location data on the CCS are equivalent the “server is adapted to send transport requests direct to at least one first target terminal.” The functions are substantially the same - to send a request from a main server (CCS) to a terminal (OCA) with instructions to download or “fill” content. The way this is accomplished is substantially the same. The claim states that the main server is adapted to send transport requests direct to at least one first target terminal over a network, and on the Open Connect Network, Netflix posts a desired manifest or emergency manifest to the CCS server that is intended for each OCA in the network. The results are substantially the same - to transmit commands from the main server/CCS server to a terminal/OCA to download content data in the manifest that needs to be retrieved. The control plane elects the specified number of OCAs as masters for a given title asset. https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 Thus, the CCS server selects OCA terminals as a download location for a given title asset. A “desired manifest” and when needed an “emergency manifest” are transport requests posted by Netflix to the CCS (Control Content Service) server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. these manifests are fetched by regularly from the CCS by each OCA terminal, which traverses the desired manifest or emergency manifest. OCAs then then query the CCS for location information files that list where each title on the desired manifest that is needed by an OCA can be downloaded. https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states a master OCA and at least one second OCA can be selected based on their relative performance: The CCS server information location list provides the address of a first server, called the “S3” server, for a download location and provides other download locations of a second and additional master OCAs: It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to all of our OCAs, so we use a tiered approach. . . . A fill escalation policy defines:

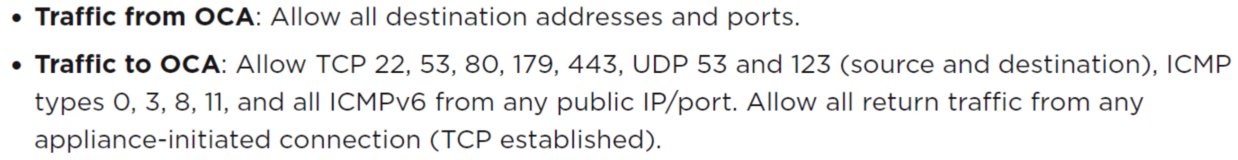
		1. How many hops away an OCA can go to download content, and how long it should wait before doing so 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so 3. Whether the OCA can go to S3, and how long it should wait before doing so The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to reach farther with less delay in order to grab that content and then share it locally with non-masters. As stated in [1e], the CCS monitors the “fill health” and performance of download locations, which is based on performance of the OCAs, to determine if that OCA will be selected as a download location or not. This desired manifest and emergency manifest along with the downloaded location information files and fill policy for master OCAs and a second (target terminal) OCA to fill from a (first terminal) OCA on the CCS are equivalent the “each such transport request includes details of data to be retrieved, the address of the first server from which the data is to be requested by the first target terminal, the addresses of at least one second target terminal to which the data from the first server to be relayed by the first target terminal.” The functions are substantially the same - to send a request from a main server (CCS) to a terminal (OCA) with instructions to download or “fill” content from specific master terminal/OCA addresses and includes an address of at least one second terminal/OCA. The way this is accomplished is substantially the same. The claim states that the main server is adapted to send transport requests that include download locations direct to at least one first target terminal over a network, and on the Open Connect Network, Netflix posts a desired manifest or emergency manifest to the CCS server along with download locations that are intended for each OCA to read on a regular basis. The results are substantially the same - to transmit commands from the main server/CCS server to a terminal/OCA to download content data in the manifest that needs to be retrieved from various addresses The main server on CCS is further adapted to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis

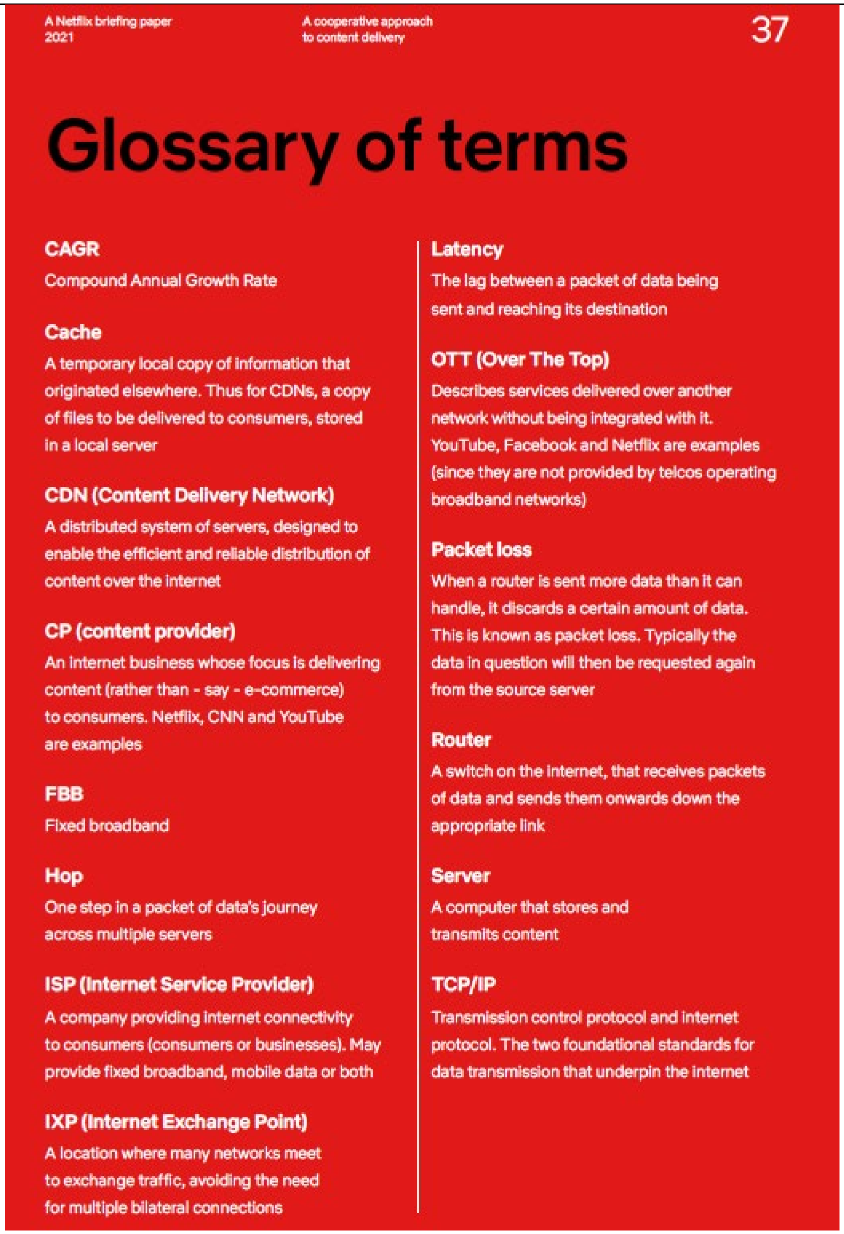
		of their relative response times. At https://netflixtechblog.com/netflix-and-fill-c43a32b490c0, Netflix wrote: The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title. The determination of the list takes into consideration several high-level factors: Title (content) availability — Does the fill source have the requested title stored? **Fill health — Can the fill source take on additional fill traffic?** A calculated route cost — Described in the next section. A fill escalation policy defines: 16. How many hops away an OCA can go to download content, and **how long it should wait before doing so** 17. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), **and how long it should wait before doing so** 18. Whether the OCA can go to S3, **and how long it should wait before doing so** (Emphasis added.) The CCS server monitors the "fill health" of each potential download locations, or fill sources that includes OCAs, and uses a fill escalation policy based upon response times to determine if the performance of the download location can take on additional traffic or not and how long the OCA should wait for a response before escalation to the next download locations. Thus, the manifest is assembled by the CCS server and download locations are assembled based on OCA performance information, and the manifest is sent to a given OCA. Therefore, the transport request is sent on a basis of said terminal performance information.
[19f]	a computer readable program code means for	OCAs can also download updates from each other – minimizing significant usage of internet "backbone" capacity during the update cycle. The OCAs work in a network to distribute updates

	causing said network terminal to modify transport requests received from said network server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server; and	among each other and to include further OCAs to which updates and content can be sent. *See* Open Connect Overview, p. 5; Fill Patterns, pp. 1-3. Netflix' OCA that are adapted to act as relay servers (see 1d above) are adapted to modify transport requests received from the main server or from other relay servers and transmit the modified transport request to selected target terminals that includes addresses of further target terminals. The CCS server will to order OCA terminals to peer or tier fill from using OCAs selected by the CCS server. The CCS server uses Appliance Section Criteria to select OCA terminals as targets in the to transfer titles from: 1) the terminal appliance that receives the most-specific route; 2) the terminal appliance that receives the route to the client's netblock with the shortest AS path; 3) the terminal appliance that receives the route to the client's netblock with the lowest multi-exit discriminator; 4) the geographically closest appliance. The CCS server includes the URL addresses of these master or target terminals in the desired manifest, which is loaded by an OCA terminal in its memory or hard drive space in order to select an OCA for downloading titles from. After receiving the desired manifest, an OCA terminal will compare the manifest with its own "actual manifest", or the list of titles currently residing with the OCA terminal. The titles missing from the actual manifest are termed the "delta" or difference between actual and desired manifest. The OCA terminal will then query the CCS terminal for a list of download locations for each title on the delta. The CCS responds, as stated in [1f] with a list of URLs that are downloadable locations of master OCAs for each individual title needed by an OCA to fill its delta: "OCAs communicate at regular intervals with the control plane services, requesting (among other things) a manifest file that contains the list of titles they should be storing and serving to members. If there is a **delta** between the list of titles in the manifest and what they are currently storing, each OCA will send a request, during its configured fill window, **that includes a list of the new or updated titles that it needs**. The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title." (Emphasis added.) *See* https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 The action of an OCA requesting download locations (master OCAs) for its delta list from the CCS, and then requesting a delta-listed title from the list of a master OCAs, is equivalent to a modified

		transport request. The OCA is using a modified list of titles (delta or missing titles list from the desired manifest) to request a title or titles from further target terminals, or master OCAs. This is equivalent to: "terminals adapted to act as relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server." The function of the two actions are substantially the same, which is to transmit to other selected relay servers or OCAs a modification of the original transport request, or desired manifest, along with URLs of those relay servers or URLs of master OCAs. The way the actions are performed are substantially the same. A delta is a modification of the manifest list, or in other words a subset list of what it is supposed to download. Instead of the terminal, or OCA, transmitting the delta list to another OCA, an "actual manifest" is sent to the CCS server which responds to the OCA with a list of URL locations to download the modified list titles of its delta list, after which the delta list is then sent to another OCA via a series of modified transfer requests. The result of these actions are substantially similar: further relay terminal addresses are sent to the OCAs, a modification of the original transport request (desired manifest) is transmitted to selected relay terminals (single or multiple delta titles to selected OCAs) are sent to addresses (URLs) of further selected relay servers (OCAs) in the form of a request for one or more titles from one or more master OCAs. Alternatively and equivalently, Netflix documentation discloses that OCA terminals, if they are clustered or if they are in the same subnet, will attempt to peer or tier fill from each other. https://openconnect.zendesk.com/hc/en-us/articles/360035618071-Fill-patterns OCA terminals in a subnet or cluster broadcast their IP and physical locations to one another and save this information. In general, appliances determine where to receive fill using selection criteria that is used by Netflix client devices. The OCA terminals then use a similar Appliance Section Criteria as the CCS server uses to select OCA terminals as targets in the subnet or cluster to transfer titles from: 1) the terminal appliance that receives the most-specific route; 2) the appliance terminal that receives the route to the client's netblock with the shortest AS path; 3) the appliance terminal that receives the route to the client's netblock with the lowest multi-exit discriminator; 4) the geographically closest terminal. The OCA terminal includes the URL addresses of these terminals

		in its memory or hard drive space in order to select an OCA for downloading titles from. See *Fill Patterns*, pp. 1-4. After receiving the desired manifest, an OCA terminal will compare the manifest with its own "actual manifest", or the list of titles currently residing with the OCA terminal. The titles missing from the actual manifest are termed the "delta" or difference between actual and desired manifest. After selecting an OCA master using the selection criteria, the OCA terminal will transmit and request the delta list items to the selected OCA(s) in the subnet or cluster in the form of download requests for each title using the URL of the target OCA. This is equivalent to: "terminals adapted to act as relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server." The function of the two actions are substantially the same, which is to transmit to other selected relay servers or OCAs a modified list of the original transport request or manifest. The way the actions are performed are substantially the same. A modified desired manifest list is a subset list called the "delta." A relay server (OCA) creates a modified transport request (delta list) of its missing titles and transports this list to another relay server (OCA sends requests to other selected OCAs for the titles on the delta list in a series of requests) and includes addresses of the other relay servers (URLs of the selected OCA terminals in the request as an address to the selected OCAs). The result of these actions are substantially similar: a modification of the original transport request (desired manifest) is transmitted to selected relay terminals (selected OCAs) with addresses (URLs) of the selected relay servers (OCAs).
[19g]	wherein data to be retrieved by said target terminals are divided into a series of packets for transmission to said target	Netflix' OCAs are adapted to communicate with the main server which is hosted by AWS. The OCA's "Report their status to the Open Connect control panel services in Amazon Web Services": The Open Connect network can make parallelized cache fill transfers. As such, it must be transferring "packets" of the file. There is evidence that each file is downloaded for cache fill in parallel in packet sizes up to 16 kilobytes.

	terminals and each of said terminals are adapted to communicate directly with said main server to acknowledge receipt of the last packet of a series routed thereto.	Additionally, the last packet of a download for a cache fill is associated with a notice to the CCS that the download has been completed. The system can also use parallel processes to perform cache transfer using TCP/IP protocols. This is only possible if the CCS is working on a different section of the file at the same time, which is packetized file transfer. There is also evidence that the cache fill transfer uses TCP/IP protocols for packetized data transfer. TCP data communication protocol, which most of the Internet included Netflix uses, requires an acknowledgement of packets when content is transmitted: TCP is a reliable byte stream delivery service which guarantees that all bytes received will be identical and in the same order as those sent. Since packet transfer by many networks is not reliable, **TCP achieves this using a technique known as positive acknowledgement with re-transmission**. This **requires the receiver to respond with an acknowledgement message as it receives the data**. The sender keeps a record of each packet it sends and maintains a timer from when the packet was sent. The sender re-transmits a packet if the timer expires before receiving the acknowledgement. The timer is needed in case a packet gets lost or corrupted.[Comer, Douglas E. (2006). Internetworking with TCP/IP: Principles, Protocols, and Architecture. Vol. 1 (5th ed.). Prentice Hall. ISBN 978-0-13-187671-2 Netflix requires network traffic to OCA be in TCP protocol:  • **Traffic from OCA**: Allow all destination addresses and ports. • **Traffic to OCA**: Allow TCP 22, 53, 80, 179, 443, UDP 53 and 123 (source and destination), ICMP types 0, 3, 8, 11, and all ICMPv6 from any public IP/port. Allow all return traffic from any appliance-initiated connection (TCP established). https://openconnect.zendesk.com/hc/en-us/articles/360035533071#routi



Glossary of terms

CAGR
Compound Annual Growth Rate

Cache
A temporary local copy of information that originated elsewhere. Thus for CDNs, a copy of files to be delivered to consumers, stored in a local server

CDN (Content Delivery Network)
A distributed system of servers, designed to enable the efficient and reliable distribution of content over the internet

CP (content provider)
An internet business whose focus is delivering content (rather than - say - e-commerce) to consumers. Netflix, CNN and YouTube are examples

FBB
Fixed broadband

Hop
One step in a packet of data's journey across multiple servers

ISP (Internet Service Provider)
A company providing internet connectivity to consumers (consumers or businesses). May provide fixed broadband, mobile data or both

IXP (Internet Exchange Point)
A location where many networks meet to exchange traffic, avoiding the need for multiple bilateral connections

Latency
The lag between a packet of data being sent and reaching its destination

OTT (Over The Top)
Describes services delivered over another network without being integrated with it. YouTube, Facebook and Netflix are examples (since they are not provided by telcos operating broadband networks)

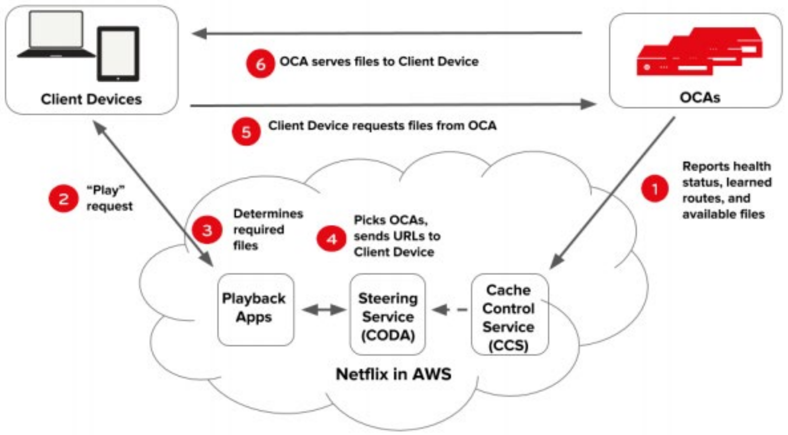
Packet loss
When a router is sent more data than it can handle, it discards a certain amount of data. This is known as packet loss. Typically the data in question will then be requested again from the source server

Router
A switch on the internet, that receives packets of data and sends them onwards down the appropriate link

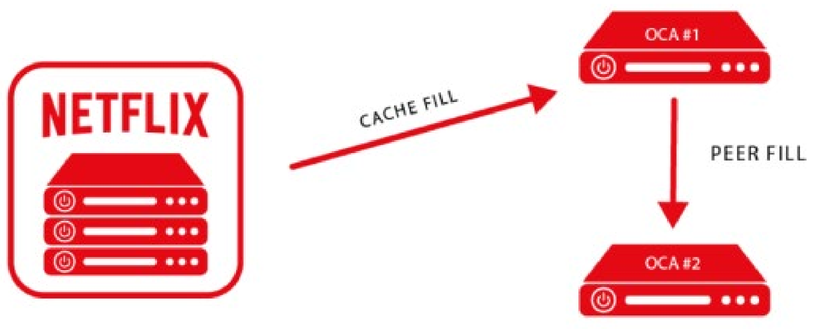
Server
A computer that stores and transmits content

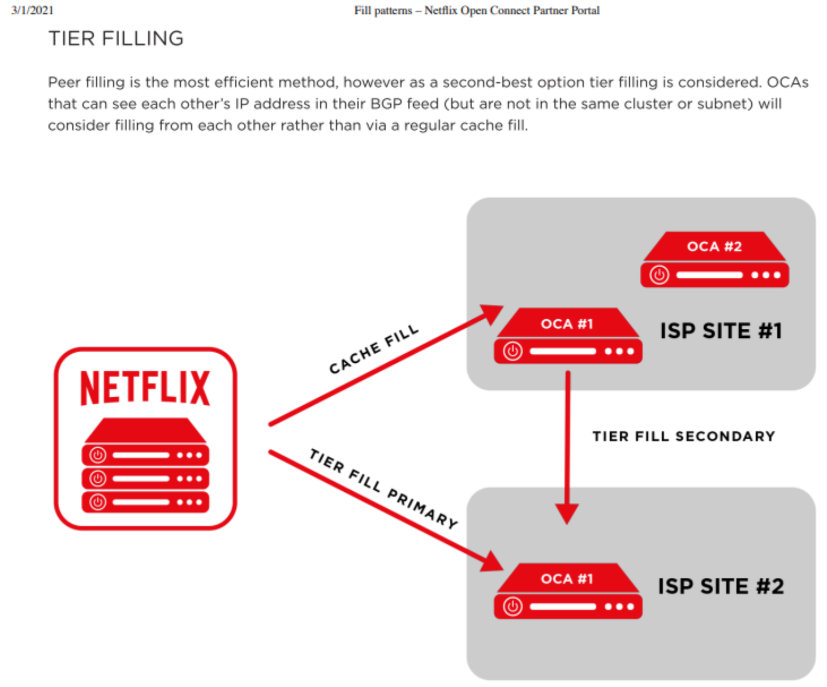
TCP/IP
Transmission control protocol and internet protocol. The two foundational standards for data transmission that underpin the internet

		"A cooperative approach to content delivery," Netflix (2021), 37. **Where does Netflix use TCP?** Netflix uses TCP for internet streaming to send packets of data for video. Additionally, Netflix specifically looks at the number of TCP connections to determine internet speeds in accordance with testing of OCAs. In another example, "After **the fill master OCA** has completed its S3 download, **it reports back to the control plane** that it now has the title stored." See https://netflixtechblog.com/netflix-and-fill-c43a32b490c0
[20a]	A computer program product for enabling a network terminal to operate as a relay server in a data communication network, the data communication network including:	Netflix uses a system called **Open Connect** to deliver Netflix TV shows and movies to members world-wide. The building blocks of Open Connect are our suite of purpose-built **server appliances, called Open Connect Appliances (OCAs)**. *See* Open Connect Overview, p. 2. These are deployed directly inside ISP networks. Netflix provides the server hardware. The OCAs report to a Open Connect control plane (CCS server) to control fill behavior (adding new files to OCAs nightly) and to compute and/or store data. *See id.* p. 3-4. Accordingly, OCAs include both an input mechanism and display mechanism.

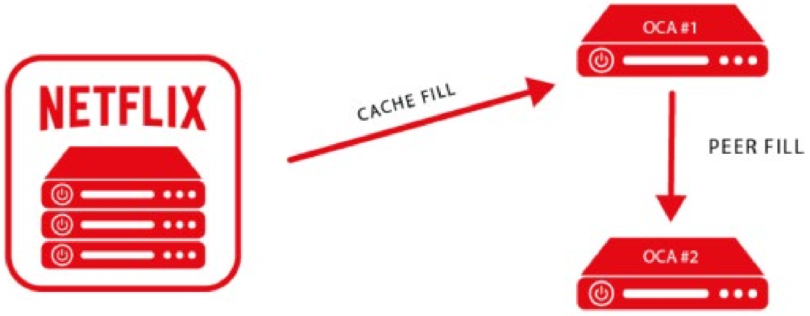
		 Client Devices OCAs 6 OCA serves files to Client Device 5 Client Device requests files from OCA 2 "Play" request 1 Reports health status, learned routes, and available files 3 Determines required files 4 Picks OCAs, sends URLs to Client Device Playback Apps Steering Service (CODA) Cache Control Service (CCS) Netflix in AWS Open Connect Appliances can be embedded in your ISP network. Embedded OCAs have the same capabilities as the OCAs that we use in our 60+ global data centers, and they are provided to qualifying ISP partners at no charge. Each embedded OCA deployment will offload a substantial amount of Netflix content traffic from peering or transport circuits. Multiple physical deployments can be distributed or clustered on a geographic or network basis to maximize local offload. Source: https://openconnect.netflix.com/en/#sample-architectures
[20b]	a plurality of terminals, a network information database, a first server from which data may be retrieved by at least one target terminal from among said	Netflix runs the operation of Open Connect from a Netflix application (CCS server) that is hosted in AWS. *See* Open Connect Overview, p. 4-5. In its global network, Netflix provides **data centers** such as an "S3" server **("first server")** housing **content ("data") on at lease one server**, and provides OCA users (such as ISP's) direct access to these data centers over the Internet that are housing the content. One or more of these **data centers** house a "**first server**" according to the claims. In deployment of Open Connect, Netflix provides Internet Service Providers with an OCA appliance direct "settlement-free interconnection (SFI)." The terminal OCA can "Connect via direct Private Network Interconnect (PNI) or IXP-based SFI peering to Netflix Open Connect Appliances **in our data centers**."

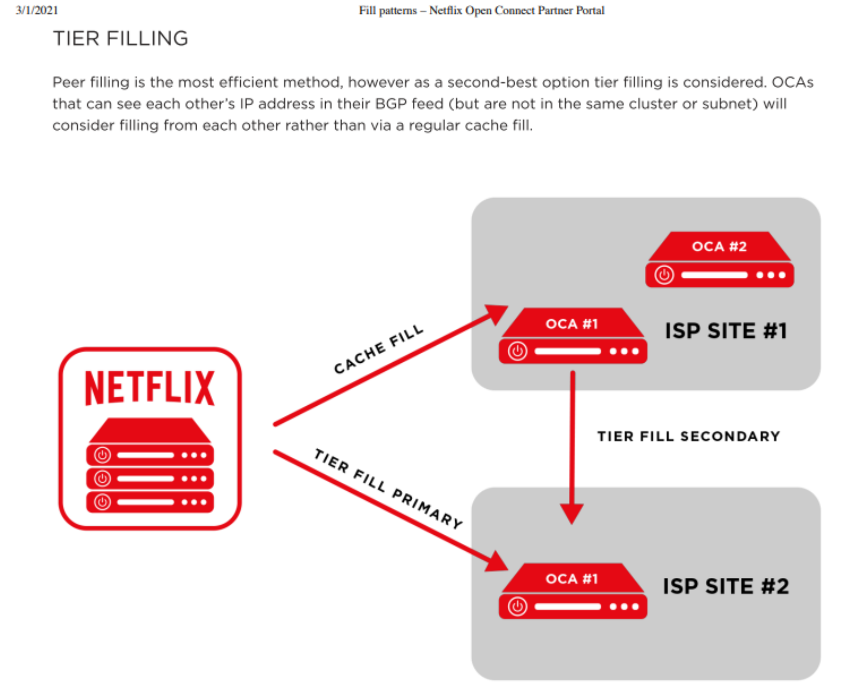
	plurality of terminals; and	“Netflix has the ability to interconnect at a number of global data center facilities and public Internet Exchange fabrics as listed on our Peering Locations page. We openly peer with any network at IXP locations where we are mutually present and we consider private interconnection as appropriate.” ISPs who do not currently participate in pubic peering might want to consider that a single IX port can support multiple peering sessions, providing **direct access to various content,** cloud, and network providers. Welcome to Open Connect, p. 3. Dkt 39 at p. 48. The following diagram also illustrates access from a target terminal (OCA #1, OCA #2) to a Netflix first server in “our data centers.” OCAs in a cluster and on the same subnet can attempt peer filling from each other. There is also Tier filling where if in different ISP sites. deploymentguide.pdf (netflix.com)

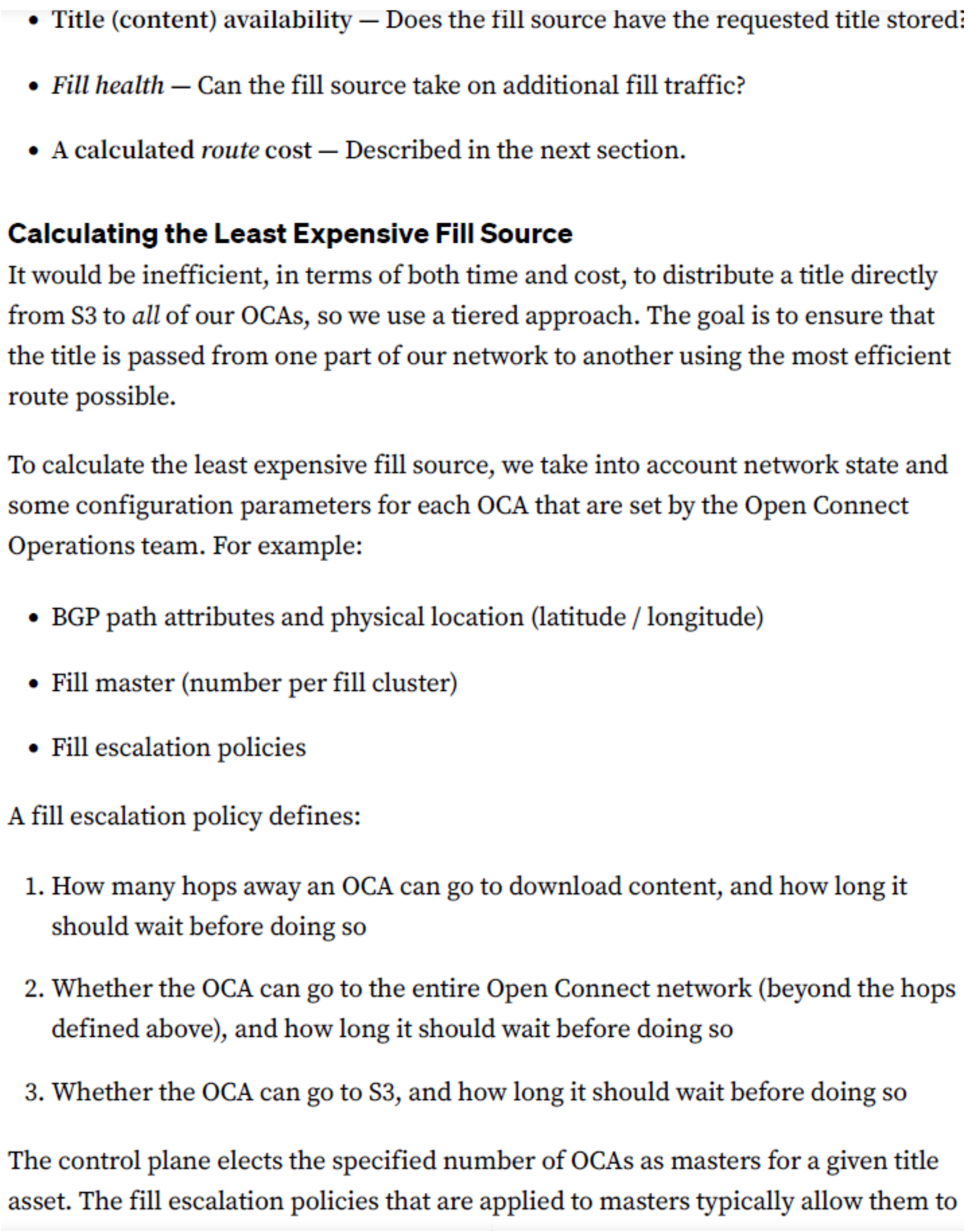
		 https://openconnect.zendesk.com/hc/en-us/articles/360035618071 2/4

		 3/1/2021 Fill patterns – Netflix Open Connect Partner Portal TIER FILLING Peer filling is the most efficient method, however as a second-best option tier filling is considered. OCAs that can see each other's IP address in their BGP feed (but are not in the same cluster or subnet) will consider filling from each other rather than via a regular cache fill. NETFLIX CACHE FILL TIER FILL PRIMARY TIER FILL SECONDARY OCA #2 OCA #1 ISP SITE #1 OCA #1 ISP SITE #2 The "target terminal selected form said plurality of terminals" language of the claim is infringed by the health and performance monitoring and OCA target terminal selection process described below in Sec. [1e].
[20c]	a main server adapted to manage selective retrieval of data from the first server by at least one target terminal selected from said plurality of terminals based	Netflix runs the operation of Open Connect from a Netflix application (main server) that is hosted in AWS. *See* Open Connect Overview, p. 4-5. In its global network, Netflix provides **data centers ("first server")** housing **content ("data") on at lease one server**, and provides OCA users (such as ISP's) direct access to these data centers over the Internet that are housing the content. One or more of these **data centers** house a "**first server**" according to the claims. In deployment of Open Connect, Netflix provides Internet Service Providers with an OCA appliance direct "settlement-free interconnection (SFI)." The terminal OCA can "Connect via direct

	on terminal performance data stored in the network information database, and wherein the main server to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis of their relative response times; said computer program product comprising:	Private Network Interconnect (PNI) or IXP-based SFI peering to Netflix Open Connect Appliances **in our data centers**." "Netflix has the ability to interconnect at a number of global data center facilities and public Internet Exchange fabrics as listed on our Peering Locations page. We openly peer with any network at IXP locations where we are mutually present and we consider private interconnection as appropriate." ISPs who do not currently participate in pubic peering might want to consider that a single IX port can support multiple peering sessions, providing **direct access to various content,** cloud, and network providers. Welcome to Open Connect, p. 3. Dkt 39 at p. 48. The following diagram also illustrates access from a target terminal (OCA #1, OCA #2) to a Netflix first server in "our data centers."

		OCAs in a cluster and on the same subnet can attempt peer filling from each other. There is also Tier filling where if in different ISP sites. deploymentguide.pdf (netflix.com)  https://openconnect.zendesk.com/hc/en-us/articles/360035618071 2/4

		 3/1/2021 Fill patterns – Netflix Open Connect Partner Portal TIER FILLING Peer filling is the most efficient method, however as a second-best option tier filling is considered. OCAs that can see each other's IP address in their BGP feed (but are not in the same cluster or subnet) will consider filling from each other rather than via a regular cache fill. NETFLIX CACHE FILL TIER FILL PRIMARY OCA #2 OCA #1 ISP SITE #1 TIER FILL SECONDARY OCA #1 ISP SITE #2 The "target terminal selected form said plurality of terminals" language of the claim is infringed by the health monitoring and OCA target terminal selection process described below in Sec. [1e]. In another example, https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states that there are "**The control plane elects the specified number of OCAs as masters. . . .**" OCAs can act as relay servers, or "masters" that target terminals can use to gain, or fill, content:

		 • Title (content) availability — Does the fill source have the requested title stored? • *Fill health* — Can the fill source take on additional fill traffic? • A calculated *route* cost — Described in the next section. **Calculating the Least Expensive Fill Source** It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to *all* of our OCAs, so we use a tiered approach. The goal is to ensure that the title is passed from one part of our network to another using the most efficient route possible. To calculate the least expensive fill source, we take into account network state and some configuration parameters for each OCA that are set by the Open Connect Operations team. For example: • BGP path attributes and physical location (latitude / longitude) • Fill master (number per fill cluster) • Fill escalation policies A fill escalation policy defines: 1. How many hops away an OCA can go to download content, and how long it should wait before doing so 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so 3. Whether the OCA can go to S3, and how long it should wait before doing so The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to

[20d]	a non-transitory computer usable medium having computer readable program code means embodied in said non-transitory medium, said computer readable program code means including:	It is confirmed by software code review that Netflix Open Connect system includes the requisite software, or computer program, to carry out the functionality of Netflix's Open Connect system.
[20e]	computer readable program code for causing said network terminal to act as relay server for serving data retrieved from said first server to at least one target terminal by receiving and responding to transport requests sent to said network terminal, each such transport request including details of data to be retrieved, the address of the first server from which	All OCA deployments are constantly monitored to ensure reliability and efficiency. Netflix **makes use of non-peak bandwidth to download the vast majority of content updates to the OCAs in network during these configurable time windows. OCAs can also download updates from each other – minimizing significant usage of internet "backbone" capacity during the update cycle.** A "desired manifest" and when needed an "emergency manifest" are posted by Netflix to the CCS server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. A "desired manifest" and when needed an "emergency manifest" are posted by Netflix to the CCS (Control Content Service) server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. Equivalently, these manifests are fetched by regularly from the CCS by each OCA terminal, which traverses the desired manifest or emergency manifest. The CCS server acts as an email inbox for manifests sent directly to the OCAs, where each OCA terminal is mandated to check regularly for manifests and download the manifests to the local OCA terminal hard drive. This desired manifest and emergency manifest and the download location data on the CCS are equivalent the "server is adapted to send transport requests direct to at least one first target

	the data is to be requested by the network terminal, the addresses of at least one second target terminal to which the data retrieved from the first server is to be relayed by the network terminal and an indication of a relative performance of a further target terminal based on the terminal performance stored in the network information database;	terminal." The functions are substantially the same - to send a request from a main server (CCS) to a terminal (OCA) with instructions to download or "fill" content. The way this is accomplished is substantially the same. The claim states that the main server is adapted to send transport requests direct to at least one first target terminal over a network, and on the Open Connect Network, Netflix posts a desired manifest or emergency manifest to the CCS server that is intended for each OCA in the network. The results are substantially the same - to transmit commands from the main server/CCS server to a terminal/OCA to download content data in the manifest that needs to be retrieved. A "desired manifest" and when needed an "emergency manifest" are transport requests posted by Netflix to the CCS (Control Content Service) server in AWS. These manifests are data structures that indicate what titles need to be transferred to a specific OCA terminal, either during an off-peak hours download window, or urgently if there is an emergent need. these manifests are fetched by regularly from the CCS by each OCA terminal, which traverses the desired manifest or emergency manifest. OCAs then then query the CCS for location information files that list where each title on the desired manifest that is needed by an OCA can be downloaded. https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 states a master OCA and at least one second OCA can be selected based on their relative performance: The CCS server information location list provides the address of a first server, called the "S3" server, for a download location and provides other download locations of a second and additional master OCAs: It would be inefficient, in terms of both time and cost, to distribute a title directly from S3 to all of our OCAs, so we use a tiered approach. . . . A fill escalation policy defines: 1. How many hops away an OCA can go to download content, and how long it should wait before doing so 2. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), and how long it should wait before doing so

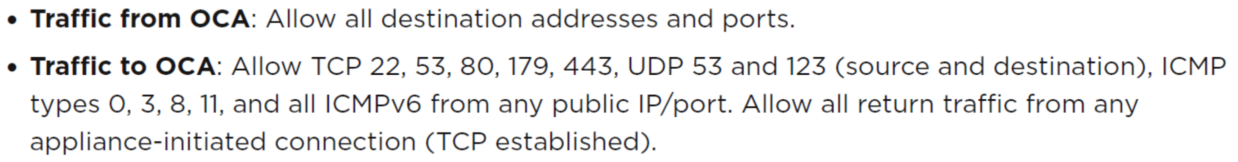
		3. Whether the OCA can go to S3, and how long it should wait before doing so The control plane elects the specified number of OCAs as masters for a given title asset. The fill escalation policies that are applied to masters typically allow them to reach farther with less delay in order to grab that content and then share it locally with non-masters. As stated in [1e], the CCS monitors the "fill health" and performance of download locations, which is based on performance of the OCAs, to determine if that OCA will be selected as a download location or not. This desired manifest and emergency manifest along with the downloaded location information files and fill policy for master OCAs and a second (target terminal) OCA to fill from a (first terminal) OCA on the CCS are equivalent the "each such transport request includes details of data to be retrieved, the address of the first server from which the data is to be requested by the first target terminal, the addresses of at least one second target terminal to which the data from the first server to be relayed by the first target terminal." The functions are substantially the same - to send a request from a main server (CCS) to a terminal (OCA) with instructions to download or "fill" content from specific master terminal/OCA addresses and includes an address of at least one second terminal/OCA. The way this is accomplished is substantially the same. The claim states that the main server is adapted to send transport requests that include download locations direct to at least one first target terminal over a network, and on the Open Connect Network, Netflix posts a desired manifest or emergency manifest to the CCS server along with download locations that are intended for each OCA to read on a regular basis. The results are substantially the same - to transmit commands from the main server/CCS server to a terminal/OCA to download content data in the manifest that needs to be retrieved from various addresses. The control plane elects the specified number of OCAs as masters for a given title asset. https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 Thus, the CCS server selects OCA terminals as a download location for a given title asset. The main server on CCS is further adapted to monitor response times of terminals in the network and in which terminals are selected to act as relay servers for a particular data transfers on the basis

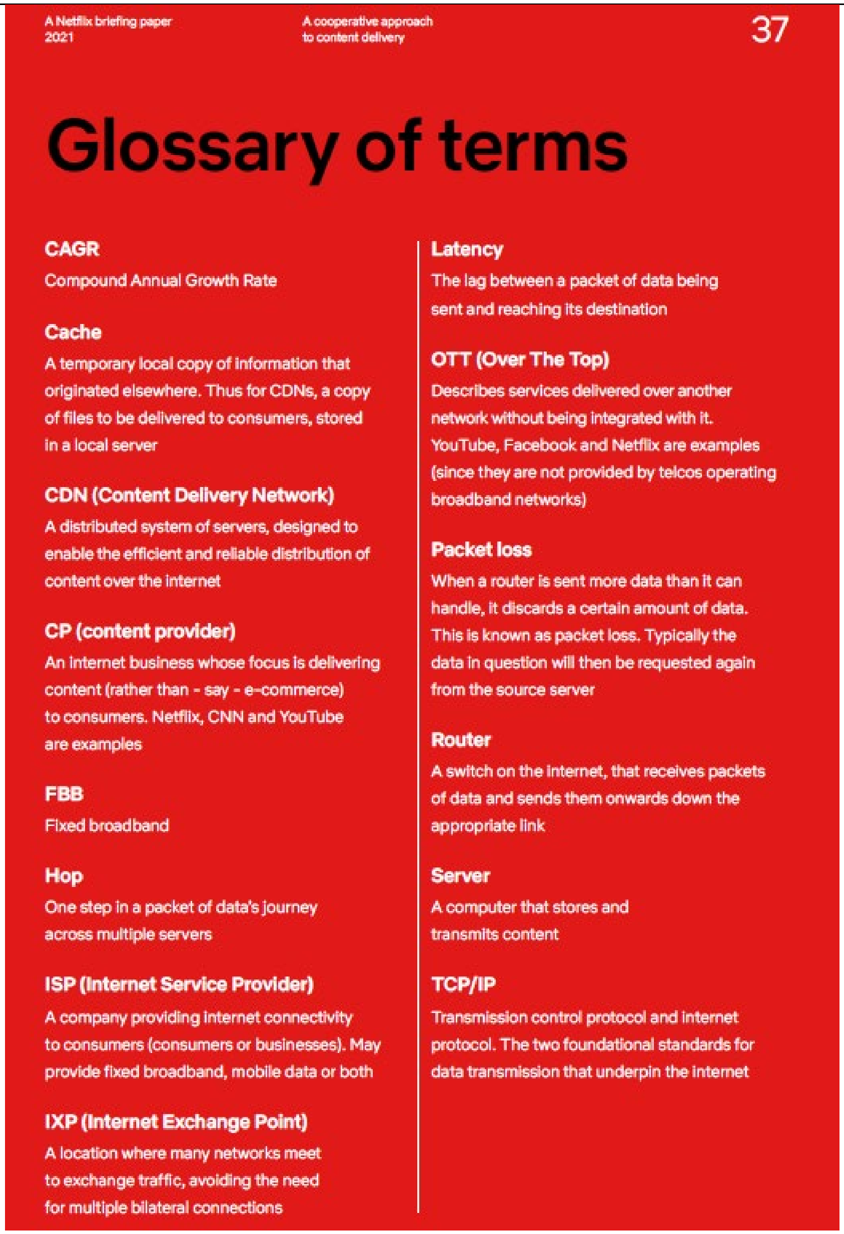
		of their relative response times. At https://netflixtechblog.com/netflix-and-fill-c43a32b490c0, Netflix wrote: The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title. The determination of the list takes into consideration several high-level factors: Title (content) availability — Does the fill source have the requested title stored? **Fill health — Can the fill source take on additional fill traffic?** A calculated route cost — Described in the next section. A fill escalation policy defines: 19. How many hops away an OCA can go to download content, and **how long it should wait before doing so** 20. Whether the OCA can go to the entire Open Connect network (beyond the hops defined above), **and how long it should wait before doing so** 21. Whether the OCA can go to S3, **and how long it should wait before doing so** (Emphasis added.) The CCS server monitors the "fill health" of each potential download locations, or fill sources that includes OCAs, and uses a fill escalation policy based upon response times to determine if the performance of the download location can take on additional traffic or not and how long the OCA should wait for a response before escalation to the next download locations. Thus, the manifest is assembled by the CCS server and download locations are assembled based on OCA performance information, and the manifest is sent to a given OCA. Therefore, the transport request is sent on a basis of said terminal performance information.
[20f]	said computer readable program code for causing	OCAs can also download updates from each other – minimizing significant usage of internet "backbone" capacity during the update cycle. The OCAs work in a network to distribute updates

	said network terminal to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server; and	among each other and to include further OCAs to which updates and content can be sent. *See* Open Connect Overview, p. 5; Fill Patterns, pp. 1-3. Netflix' OCA that are adapted to act as relay servers (see 1d above) are adapted to modify transport requests received from the main server or from other relay servers and transmit the modified transport request to selected target terminals that includes addresses of further target terminals. The CCS server will to order OCA terminals to peer or tier fill from using OCAs selected by the CCS server. The CCS server uses Appliance Section Criteria to select OCA terminals as targets in the to transfer titles from: 1) the terminal appliance that receives the most-specific route; 2) the terminal appliance that receives the route to the client's netblock with the shortest AS path; 3) the terminal appliance that receives the route to the client's netblock with the lowest multi-exit discriminator; 4) the geographically closest appliance. The CCS server includes the URL addresses of these master or target terminals in the desired manifest, which is loaded by an OCA terminal in its memory or hard drive space in order to select an OCA for downloading titles from. After receiving the desired manifest, an OCA terminal will compare the manifest with its own "actual manifest", or the list of titles currently residing with the OCA terminal. The titles missing from the actual manifest are termed the "delta" or difference between actual and desired manifest. The OCA terminal will then query the CCS terminal for a list of download locations for each title on the delta. The CCS responds, as stated in [1f] with a list of URLs that are downloadable locations of master OCAs for each individual title needed by an OCA to fill its delta: "OCAs communicate at regular intervals with the control plane services, requesting (among other things) a manifest file that contains the list of titles they should be storing and serving to members. If there is a **delta** between the list of titles in the manifest and what they are currently storing, each OCA will send a request, during its configured fill window, **that includes a list of the new or updated titles that it needs**. The response from the control plane in AWS is a ranked list of potential download locations, a.k.a. fill sources, for each title." (Emphasis added.) *See* https://netflixtechblog.com/netflix-and-fill-c43a32b490c0 The action of an OCA requesting download locations (master OCAs) for its delta list from the CCS, and then requesting a delta-listed title from the list of a master OCAs, is equivalent to a modified

		transport request. The OCA is using a modified list of titles (delta or missing titles list from the desired manifest) to request a title or titles from further target terminals, or master OCAs. This is equivalent to: “terminals adapted to act as relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server.” The function of the two actions are substantially the same, which is to transmit to other selected relay servers or OCAs a modification of the original transport request, or desired manifest, along with URLs of those relay servers or URLs of master OCAs. The way the actions are performed are substantially the same. A delta is a modification of the manifest list, or in other words a subset list of what it is supposed to download. Instead of the terminal, or OCA, transmitting the delta list to another OCA, an “actual manifest” is sent to the CCS server which responds to the OCA with a list of URL locations to download the modified list titles of its delta list, after which the delta list is then sent to another OCA via a series of modified transfer requests. The result of these actions are substantially similar: further relay terminal addresses are sent to the OCAs, a modification of the original transport request (desired manifest) is transmitted to selected relay terminals (single or multiple delta titles to selected OCAs) are sent to addresses (URLs) of further selected relay servers (OCAs) in the form of a request for one or more titles from one or more master OCAs. Alternatively and equivalently, Netflix documentation discloses that OCA terminals, if they are clustered or if they are in the same subnet, will attempt to peer or tier fill from each other. https://openconnect.zendesk.com/hc/en-us/articles/360035618071-Fill-patterns OCA terminals in a subnet or cluster broadcast their IP and physical locations to one another and save this information. In general, appliances determine where to receive fill using selection criteria that is used by Netflix client devices. The OCA terminals then use a similar Appliance Section Criteria as the CCS server uses to select OCA terminals as targets in the subnet or cluster to transfer titles from: 1) the terminal appliance that receives the most-specific route; 2) the appliance terminal that receives the route to the client’s netblock with the shortest AS path; 3) the appliance terminal that receives the route to the client’s netblock with the lowest multi-exit discriminator; 4) the geographically closest terminal. The OCA terminal includes the URL addresses of these terminals

		in its memory or hard drive space in order to select an OCA for downloading titles from. See *Fill Patterns*, pp. 1-4. After receiving the desired manifest, an OCA terminal will compare the manifest with its own “actual manifest”, or the list of titles currently residing with the OCA terminal. The titles missing from the actual manifest are termed the “delta” or difference between actual and desired manifest. After selecting an OCA master using the selection criteria, the OCA terminal will transmit and request the delta list items to the selected OCA(s) in the subnet or cluster in the form of download requests for each title using the URL of the target OCA. This is equivalent to: “terminals adapted to act as relay servers are adapted to modify transport requests received from the main server or from other relay servers and to transmit the modified transport request to selected target terminals from a set of target terminals identified in the transport request, wherein the modified transport request further includes addresses of further target terminals for which the recipient of the modified transport request is to act as relay server.” The function of the two actions are substantially the same, which is to transmit to other selected relay servers or OCAs a modified list of the original transport request or manifest. The way the actions are performed are substantially the same. A modified desired manifest list is a subset list called the “delta.” A relay server (OCA) creates a modified transport request (delta list) of its missing titles and transports this list to another relay server (OCA sends requests to other selected OCAs for the titles on the delta list in a series of requests) and includes addresses of the other relay servers (URLs of the selected OCA terminals in the request as an address to the selected OCAs). The result of these actions are substantially similar: a modification of the original transport request (desired manifest) is transmitted to selected relay terminals (selected OCAs) with addresses (URLs) of the selected relay servers (OCAs).
[20g]	wherein data to be retrieved by said target terminals are divided into a series of packets for transmission to said target	Netflix’ OCAs are adapted to communicate with the main server which is hosted by AWS. The OCA’s “Report their status to the Open Connect control panel services in Amazon Web Services”: The Open Connect network can make parallelized cache fill transfers. As such, it must be transferring “packets” of the file. There is evidence that each file is downloaded for cache fill in parallel in packet sizes up to 16 kilobytes.

	terminals and each of said terminals are adapted to communicate directly with said main server to acknowledge receipt of the last packet of a series routed thereto.	Additionally, the last packet of a download for a cache fill is associated with a notice to the CCS that the download has been completed. The system can also use parallel processes to perform cache transfer using TCP/IP protocols. This is only possible if the CCS is working on a different section of the file at the same time, which is packetized file transfer. There is also evidence that the cache fill transfer uses TCP/IP protocols for packetized data transfer. TCP data communication protocol, which most of the Internet included Netflix uses, requires an acknowledgement of packets when content is transmitted: TCP is a reliable byte stream delivery service which guarantees that all bytes received will be identical and in the same order as those sent. Since packet transfer by many networks is not reliable, **TCP achieves this using a technique known as positive acknowledgement with re-transmission**. This **requires the receiver to respond with an acknowledgement message as it receives the data**. The sender keeps a record of each packet it sends and maintains a timer from when the packet was sent. The sender re-transmits a packet if the timer expires before receiving the acknowledgement. The timer is needed in case a packet gets lost or corrupted.[Comer, Douglas E. (2006). Internetworking with TCP/IP: Principles, Protocols, and Architecture. Vol. 1 (5th ed.). Prentice Hall. ISBN 978-0-13-187671-2 Netflix requires network traffic to OCA be in TCP protocol:  • **Traffic from OCA**: Allow all destination addresses and ports. • **Traffic to OCA**: Allow TCP 22, 53, 80, 179, 443, UDP 53 and 123 (source and destination), ICMP types 0, 3, 8, 11, and all ICMPv6 from any public IP/port. Allow all return traffic from any appliance-initiated connection (TCP established). https://openconnect.zendesk.com/hc/en-us/articles/360035533071#routi



Glossary of terms

CAGR
Compound Annual Growth Rate

Cache
A temporary local copy of information that originated elsewhere. Thus for CDNs, a copy of files to be delivered to consumers, stored in a local server

CDN (Content Delivery Network)
A distributed system of servers, designed to enable the efficient and reliable distribution of content over the internet

CP (content provider)
An internet business whose focus is delivering content (rather than - say - e-commerce) to consumers. Netflix, CNN and YouTube are examples

FBB
Fixed broadband

Hop
One step in a packet of data's journey across multiple servers

ISP (Internet Service Provider)
A company providing internet connectivity to consumers (consumers or businesses). May provide fixed broadband, mobile data or both

IXP (Internet Exchange Point)
A location where many networks meet to exchange traffic, avoiding the need for multiple bilateral connections

Latency
The lag between a packet of data being sent and reaching its destination

OTT (Over The Top)
Describes services delivered over another network without being integrated with it. YouTube, Facebook and Netflix are examples (since they are not provided by telcos operating broadband networks)

Packet loss
When a router is sent more data than it can handle, it discards a certain amount of data. This is known as packet loss. Typically the data in question will then be requested again from the source server

Router
A switch on the internet, that receives packets of data and sends them onwards down the appropriate link

Server
A computer that stores and transmits content

TCP/IP
Transmission control protocol and internet protocol. The two foundational standards for data transmission that underpin the internet

		“A cooperative approach to content delivery,” Netflix (2021), 37. Where does Netflix use TCP? Netflix uses TCP for internet streaming to send packets of data for video. Additionally, Netflix specifically looks at the number of TCP connections to determine internet speeds in accordance with testing of OCAs. In another example, “After **the fill master OCA** has completed its S3 download, **it reports back to the control plane** that it now has the title stored.” See https://netflixtechblog.com/netflix-and-fill-c43a32b490c0